

**ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA**

---

**DIPARTIMENTO DI INGEGNERIA E ARCHITETTURA**

**CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA**

**TESI DI LAUREA MAGISTRALE IN DATA MINING**

**ANALISI E SVILUPPO DI UN'APPLICAZIONE PER IL  
MONITORAGGIO REAL-TIME DI SISTEMI BUSINESS CRITICAL  
MEDIANTE PIATTAFORMA ORACLE APEX**

**RELATORE**

Prof. Claudio Sartori

**CANDIDATA**

Nicoletta Recchia

**CORRELATORE**

T. A. Manager Simone Morassutto

**ANNO ACCADEMICO 2021/2022**

*A Sandro ed Anna Rita.*

# Indice

<b>Abstract</b> .....	<b>5</b>
<b>Oracle Application Express (APEX)</b> .....	<b>7</b>
<b>1.1 Application Development nelle aziende</b> .....	<b>7</b>
<b>1.2 Low Code Development</b> .....	<b>7</b>
<b>1.3 APEX</b> .....	<b>8</b>
<b>1.4 Soluzioni Offerte da Oracle APEX</b> .....	<b>11</b>
1.4.1 Applicazioni Cloud e Mobile .....	11
1.4.2 Sostituzione dei fogli di calcolo .....	12
1.4.3 Condivisione di dati esterni .....	12
1.4.4 Report Interattivi .....	13
<b>1.5 Funzionalità</b> .....	<b>13</b>
1.5.1 Dati e Reporting .....	13
1.5.2 Ambiente Mobile.....	15
1.5.3 Universal Theme .....	16
1.5.4 Autenticazione e Autorizzazione.....	17
<b>1.6 Oracle Application Express Development Environment</b> .....	<b>17</b>
1.6.1 Application Builder .....	18
1.6.2 SQL Workshop.....	19
1.6.3 Team Development e Gallery .....	23
<b>1.7 Anatomia di un'applicazione Oracle APEX</b> .....	<b>24</b>
1.7.1 Componenti di un'applicazione.....	25
1.7.2 Page Designer.....	26
<b>1.8 Creare un'applicazione APEX</b> .....	<b>32</b>
<b>AMS: Real-Time Incidents</b> .....	<b>37</b>
<b>2.1 Analisi dei requisiti</b> .....	<b>37</b>
<b>2.2 Home Page Panoramica</b> .....	<b>39</b>
2.2.1 Mission Critical .....	41
2.2.2 Production .....	46
2.2.3 Development Test.....	47
2.2.4 Altro .....	48
2.2.5 Eventi Pinnati .....	53
2.2.6 Ultimi Eventi .....	61
<b>2.3 Incident</b> .....	<b>62</b>
<b>2.4 Cloud Control</b> .....	<b>70</b>
<b>2.5 Clienti</b> .....	<b>71</b>
<b>Sviluppi Futuri</b> .....	<b>75</b>
<b>3.1 Mail Support</b> .....	<b>75</b>

<b>3.2 Servizio Ticket .....</b>	<b>75</b>
<b>3.3 Invia Incident su Telegram .....</b>	<b>76</b>
<b>3.4 Bacheca.....</b>	<b>76</b>
<b>3.5 Calendario .....</b>	<b>76</b>
<b>3.6 Centralino.....</b>	<b>76</b>
<b>Conclusioni.....</b>	<b>78</b>

# Abstract

Il presente elaborato di tesi intende offrire al lettore una accurata esposizione delle attività di analisi e realizzazione del progetto di tirocinio, svolto presso l'azienda Mediamente Consulting, inerente all'implementazione di un sistema di anomaly detection/monitoraggio real-time che consente la visualizzazione grafica di alert/incident, provenienti da server business-critical costantemente osservati, dando particolare enfasi al loro livello di criticità.

L'applicazione è stata sviluppata usufruendo della più diffusa piattaforma low code per il development di applicazioni web in ambito business enterprise, Oracle Application Express (APEX). L'utilizzo di tale strumento ha consentito l'ampliamento di conoscenze pregresse, quali il linguaggio di interrogazione SQL (Structured Query Language), frequentemente impiegato per l'interazione con Database Management Systems (DBMS); il linguaggio di programmazione procedurale PL/SQL funzionale anch'esso all'interrogazione di database; i linguaggi HTML, CSS e JavaScript; requisiti tutti indispensabili, seppur in diversa misura, per un APEX developer.

Di seguito verrà brevemente descritto il contenuto di ciascuno dei capitoli del documento di tesi:

- I. il primo capitolo è interamente dedicato all'introduzione di Oracle APEX ponendo, dapprima, particolare attenzione alla sua natura di low code, web-based, declarative e data centric development tool e alla sua architettura a tre livelli. In secondo luogo, verranno esposte le soluzioni e le molteplici funzionalità offerte da APEX, i principali casi d'uso per cui viene impiegato, per poi fornire una panoramica dell'ambiente di sviluppo (IDE) ed i suoi fondamentali componenti. Si procederà quindi a descrivere l'anatomia di una applicazione APEX che risiede all'interno di una precisa area di lavoro, denominata workspace. Per concludere, verrà mostrato come creare una semplice applicazione.
- II. Il secondo capitolo è riservato all'analisi dei requisiti delineati in fase di progettazione del sistema di monitoraggio AMS: Real-Time Incidents, esplorandone le funzionalità ed i tool, i linguaggi di programmazione, i maggiori componenti APEX per la reportistica e la visualizzazione richiesti. Successivamente, verrà presentata ciascuna delle sezioni dell'applicazione con approfondimenti relativi ai

tool APEX utilizzati per l'implementazione e le modalità di interrogazione dei dati relative alle singole aree.

- III. Il terzo ed ultimo capitolo raccoglie alcune osservazioni conclusive e propone possibili sviluppi futuri dell'applicazione oggetto del presente studio.

# Capitolo 1

## Oracle Application Express (APEX)

### 1.1 Application Development nelle aziende

Le maggiori criticità legate allo sviluppo di applicazioni nel mondo enterprise riguardano i costi e tempi di sviluppo, la mancata efficace collaborazione tra le figure aziendali e l'assenza di procedure di backlog. Ogni linea di codice è associata ad un costo per la scrittura, il debugging, la review, il tuning e le modifiche necessarie. Maggiore è il task da realizzare, maggiori saranno le risorse ad esso destinate ed i conseguenti costi. Questi ultimi, peraltro, non potranno essere ridotti mediante l'assegnazione dell'intero onere ad un full stack developer. In molti casi, tempi di sviluppo, cambiamenti ed esigenze del business non avanzano simultaneamente. Come sopra accennato, la ridotta collaborazione tra i soggetti coinvolti quali sviluppatori, project manager, business user ed end-users, ostacola l'ottimizzazione dei tempi di implementazione delle applicazioni. Inoltre, le pratiche di backlog raramente vengono attuate a fronte del mancato impiego di risorse ad esso dedicate, venendo così a mancare una pronta analisi delle eventuali problematiche e l'elaborazione di strategie di risoluzione delle stesse.

### 1.2 Low Code Development

Le complessità sopra elencate, possono trovare una naturale soluzione mediante l'adozione di piattaforme di sviluppo low code. Tali strumenti, dotati di interfaccia grafica di estrema semplicità, forniscono un ambiente di sviluppo in grado di creare applicazioni software che richiedono minore codifica rispetto ai tradizionali framework di programmazione. Essi, inoltre, si distinguono per le migliori funzionalità visive e la compatibilità con dispositivi desktop e mobile. Appare pertanto opportuno sottolineare i principali benefici che possono trarsi dall'utilizzo di tool low coding quali, la facilità di utilizzo, la crescente produttività dovuta alla rapidità di implementazione rispetto alle tradizionali tecnologie, l'accessibilità non riservata al full stack developer, ma fruibile da un ampio pool di risorse. Da sottolineare sono poi gli ulteriori vantaggi in termini di scalabilità, dovuta alla possibilità di creare un'app low code con funzionalità base per poi accrescerne la complessità; di estensibilità, in quanto i mutevoli

requisiti di business aziendali possono velocemente essere tradotti e integrati all'interno dell'applicazione; infine, la riduzione dei costi di development.

Tutti i benefits sopra citati facilitano le prestazioni degli sviluppatori, i quali non sono vincolati a focalizzare l'attenzione su problematiche come l'interazione con i database, ma piuttosto si impegnano nella risoluzione dei problemi di business aziendali e nel raggiungimento degli obiettivi di business.

## 1.3 APEX

Oracle APEX è la *“piattaforma di sviluppo di applicazioni low code enterprise più diffusa al mondo (...) consente di creare applicazioni enterprise scalabili e sicure, con funzioni di primissimo livello, che possono essere distribuite dovunque: nel cloud e in locale.”*<sup>1</sup>

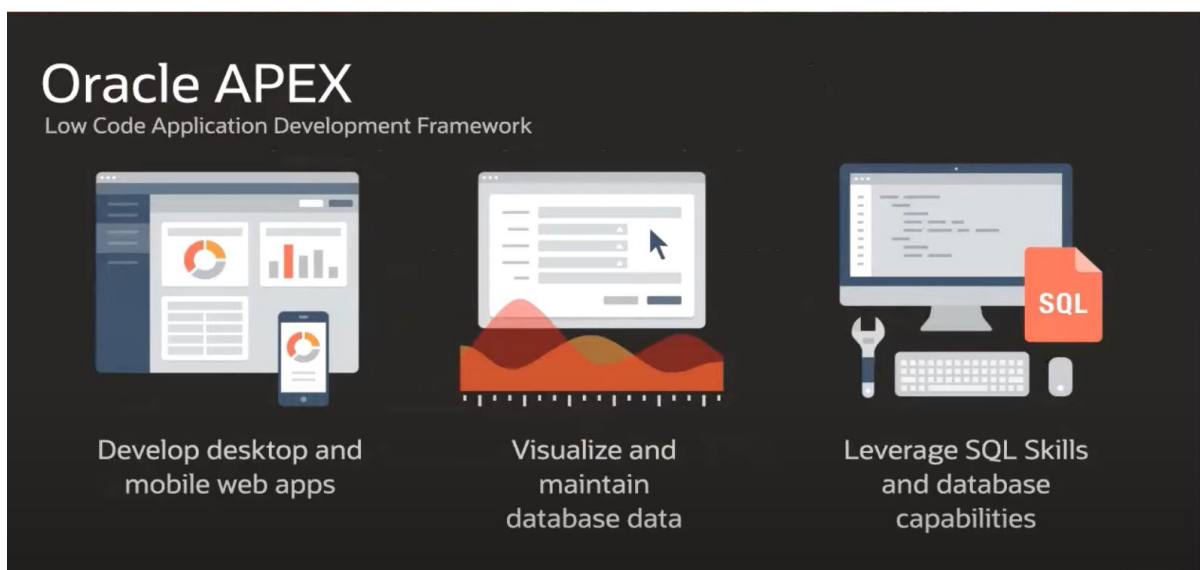


FIGURA 1: ORACLE APPLICATION EXPRESS (APEX)<sup>2</sup>

APEX consente di sviluppare desktop e mobile enterprise web app inerenti alla visualizzazione, manipolazione e storage di dati, il tutto possedendo skill SQL e conoscenze sui database e con l'ulteriore vantaggio della minimizzazione del codice.

---

<sup>1</sup> <https://apex.oracle.com/it/>

<sup>2</sup> <https://apex.oracle.com/it/>



È un browser-based Rapid Application Development (RAD) tool che agevola la creazione di applicazioni web moderne e data-centric, realizzabili con estrema rapidità e con significativo minore dispendio di tempi e risorse nelle attività di coding, a differenza delle tradizionali tecnologie e tecniche di programmazione. Infatti, un developer può garantire un prodotto finale enterprise 20 volte più velocemente e con numerosità di codice 100 volte inferiore; è in grado di risolvere problemi real-time, fornendo soluzioni che non richiedono molte competenze pregresse per la concretizzazione dei requisiti di business. Nello specifico, si parla di applicazioni web realizzate ed eseguite su database Oracle, che possono includere report, form, dashboard, grafici, calendari ed altri componenti funzionali alla visualizzazione e manipolazione di dati. Inoltre, ciascuna sezione può essere raggiunta creando semplici percorsi di navigazione e può essere agevolmente personalizzata in modo tale da rispecchiare appieno gli obiettivi aziendali prefissati.

Dalle definizioni appena ricordate, possono essere pertanto enumerate le differenti e singolari proprietà che distinguono tale framework - funzionale al developing di web enterprise app - dagli abituali strumenti per l'implementazione di applicazioni web.

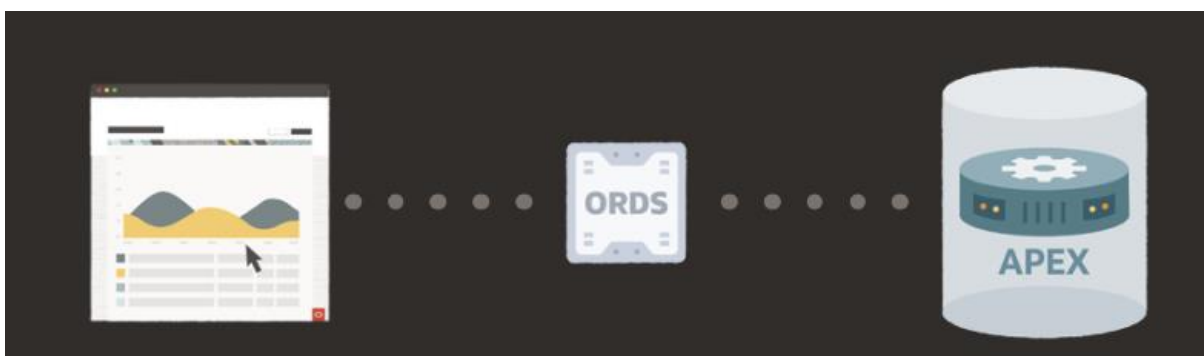
In primo luogo, APEX è un ambiente di sviluppo dichiarativo che accelera il developing e la distribuzione di applicazioni web data-centric. In qualità di framework dichiarativo nessun codice viene generato o compilato e risponde al paradigma "Rapidly Develop, Customize and Deliver", ossia garantisce rapidamente soluzioni personalizzate anche mediante procedure guidate (built-in wizards). Tali wizards, facilitano la creazione delle differenti pagine di cui si comporrà un'applicazione rendendole conformi a quelli che sono gli obiettivi aziendali. Tra le principali funzioni predefinite troviamo: un'interfaccia utente, controlli di navigazione, metodi di gestione delle form, page rendering e database utilities che indirizzano l'utente durante tutto il processo di generazione della soluzione finale, tramutando un prototipo in prodotto finale rapidamente e garantendo l'accesso, nonché l'interazione con il database, in totale sicurezza. Ogni dato dell'applicazione viene infatti immagazzinato nelle tabelle del database Oracle come un metadato che può essere migliorato utilizzando query SQL o PL/SQL personalizzate. Infine, quando l'applicazione viene eseguita, il motore di APEX legge tali metadati per poi mostrare il risultato.

Un secondo aspetto da sottolineare è la sua natura di piattaforma nativa low code, i cui vantaggi sono appunto la velocità nella application delivery, dovuta soprattutto alla minima generazione di codice rispetto al coding tradizionale richiesto per le applicazioni web.

Tra le peculiarità di APEX, è necessario evidenziarne le facoltà di web browser based development tool per desktop e mobile web app. La proprietà web-based si riscontra sia nella fase di sviluppo che nell'esecuzione delle applicazioni e, in entrambi i casi, ne consegue che non è necessario ricorrere ad alcun client software. L'IDE (Integrated Development Environment) è un web browser e sullo stesso vengono anche eseguite le applicazioni; pertanto, gli unici requisiti necessari all'utilizzo del tool sono un browser e la connessione ad una rete Internet o Intranet. Una applicazione è infatti costituita da una o molteplici web pages che includono report, forms, grafici ed altri componenti volti a rappresentare le logiche di business; ciascuna delle pagine viene tradotta dall'APEX engine in codice HTML lato client e in codice SQL o PL/SQL lato server. L'ambiente supporta inoltre linguaggi di programmazione per app web come CSS e JavaScript.

Un'ulteriore fondamentale caratteristica di APEX è la sua competenza nella costruzione di applicazioni data-centric. L'utilizzo di tale tool risulta essere ottimale se gli obiettivi di business si traducono nel designing di applicazioni che consentono lo storage, manipolazione, restituzione e visualizzazione dei dati da un database Oracle. Il motivo per cui APEX può vantare tali benefici è, senz'altro, dovuto alla sua qualità di piattaforma nativa per il database Oracle. Disporre di un database Oracle implica il poter usufruire di APEX e, di conseguenza, l'utilizzo di tale framework rende possibile accedere in modo nativo ad un database Oracle ed ereditarne tutte le funzionalità e features, quali l'utilizzo di packages PL/SQL, la sicurezza, le elevate performance, la robustezza, scalabilità ed altro ancora.

È opportuno riporre l'attenzione anche sulla semplice architettura di cui APEX dispone, ovvero una architettura a tre livelli (3-Tier Architecture).



**FIGURA 2: ORACLE APEX 3-TIER ARCHITECTURE<sup>3</sup>**

<sup>3</sup> <https://apex.oracle.com/it/platform/architecture/>

Come è intuibile dalla figura illustrata sopra, il primo livello è costituito dal browser web, il livello centrale da un server web, mentre l'ultimo livello è riservato al database. La logica dell'architettura prevede che le richieste del Browser vengano indirizzate al database tramite un web server, nello specifico mediante Oracle REST Data Service (ORDS). Qualsiasi attività di business logic, processamento e manipolazione di dati, viene eseguita all'interno del database, garantendo accesso ai dati con tempi di latenza pari a zero e massime prestazioni. All'interno del database, la richiesta viene elaborata da Oracle APEX engine e, al suo completamento, il risultato verrà inviato tramite i servizi REST al browser. Più precisamente, quando viene creata o modificata un app esistente, APEX rispettivamente crea o modifica dati memorizzati nelle table del database; nel momento in cui verrà eseguita, APEX andrà a leggere i metadati presenti sul database e visualizzerà la pagina oppure elaborerà le richieste relative a quella pagina. L'elaborazione avviene tramite queries SQL direttamente sui database schemas. Infine, come sopra già accennato, è necessario sottolineare che tale framework richiede pochi requisiti di utilizzo, ovvero limitate conoscenze base dei linguaggi SQL e PL/SQL, HTML, JavaScript e CSS.

## **1.4 Soluzioni Offerte da Oracle APEX**

Sono diversi i casi d'uso attraverso i quali avvalersi delle numerose funzionalità offerte da un tool come APEX; tra i principali troviamo lo sviluppo di nuove applicazioni cloud e mobile, la sostituzione dei fogli di calcolo, la condivisione di dati esterni con partner e clienti aziendali e la creazione di report interattivi.

### **1.4.1 Applicazioni Cloud e Mobile**

Uno dei maggiori casi d'uso consiste nel creare nuove applicazioni che soddisfino con maggiore tempestività i mutevoli propositi aziendali. Gli ambienti enterprise sono dotati di numerose applicazioni che necessitano di essere costantemente revisionate, aggiornate e migliorate al fine di ottenere un vantaggio competitivo; tuttavia, l'assenza di procedure di backlog e la fruizione di tecniche e software di programmazione tradizionali non accelerano il processo di raggiungimento degli obiettivi di business. Le applicazioni in questione non sono di tipo enterprise a lunga durata, ma occorrono per un breve arco temporale compatibilmente con i

variabili requisiti aziendali. APEX, in qualità di Rapid Application Development (RAD) framework, risponde alle esigenze sopra citate mediante la creazione di applicazioni in pochi giorni o settimane, rendendo possibile apportare modifiche in real-time e permettendo il confronto diretto con gli utenti finali. I tempi di sviluppo sono notevolmente ridotti, consentendo di creare un prototipo iniziale in pochi minuti, grazie alle proprietà di low coding e alle componenti APEX con un'interfaccia intuitiva e di estrema semplicità d'utilizzo. Inoltre, è utile sottolineare che Oracle APEX è disponibile senza costi aggiuntivi se si dispone di una licenza per il database Oracle.

### **1.4.2 Sostituzione dei fogli di calcolo**

L'esigenza di sostituire i comuni spreadsheets con forme di visualizzazione e gestione dati dal design moderno ed intuitivo, nasce principalmente a causa della generazione di duplicati dello stesso foglio di calcolo, ma con dati differenti nei vari dipartimenti aziendali. Questa complicazione è dovuta soprattutto all'estrema semplicità di creazione di tali fogli da parte delle risorse aziendali coinvolte; infatti, una volta che un utente dispone dei dati oggetto d'analisi, utilizzerà un foglio di calcolo per la creazione di report, che verrà certamente acceduto e modificato da un altro utente, generando duplicati e peggiorando le performance dei processi di analisi aziendale. Invece, le procedure guidate di cui APEX è dotato, consentono, con rapidità, la trasformazione dei fogli di calcolo in una tabella dati nel database Oracle e la conseguente visualizzazione e gestione dei dati mediante numerose varianti di report, form ed altre componenti. Nello specifico, è sufficiente caricare o trascinare uno spreadsheet XLS, CSV, XML o JASON il quale verrà così tramutato in una table. Il vantaggio è evidente: l'applicazione creata diverrà l'unico punto di accesso ai dati mediante un semplice URL e ciascuna modifica apportata verrà immagazzinata nelle tabelle, evitando modifiche e duplicazioni indesiderate.

### **1.4.3 Condivisione di dati esterni**

Un ulteriore caso d'uso, per cui il tool APEX rappresenta una vera e propria soluzione, riguarda la condivisione di dati con utenti esterni all'azienda, i partner e clienti. I benefici che tale piattaforma può vantare e le soluzioni che può offrire, possono giovare non solo all'organizzazione interna dell'azienda, ma anche accrescere le relazioni e la soddisfazione di figure quali clienti e partner. Questi, infatti, avranno la possibilità di interagire con un prodotto altamente personalizzato, sì da essere incentivati a richiederne di ulteriori in relazione alle diverse esigenze aziendali.

## **1.4.4 Report Interattivi**

Altre criticità risolvibili tramite APEX riguardano i sistemi ERP (Enterprise Resource Planning) e la gestione di dati eterogenei provenienti da numerosi ERP.

Spesso, tali sistemi non forniscono report di cui si necessita davvero, potrebbero mancare di alcune funzionalità e richiedere processi che impiegano del tempo. Una soluzione efficace potrebbe essere la creazione di estensioni degli ERP o dei vari software aziendali con Oracle APEX al fine di migliorarne sensibilmente la produttività. Per quanto riguarda, invece, i dati provenienti da diverse fonti ERP possono frequentemente causare la produzione di report che non soddisfano i requisiti, non fornendo soluzioni adeguate ai processi decisionali aziendali ed, altresì, non garantendo controlli sulla sicurezza, permettendo così ad utenti non autorizzati di accedere a dati riservati. Usufruento delle diverse funzionalità di APEX, è possibile migliorare e velocizzare i processi di acquisizione ed integrazione di dati da parte di fonti eterogenee; inoltre, mediante le tipologie di reporting e data visualization tool proposte da APEX, è possibile creare dashboard con report interattivi ed impostare controlli sulla sicurezza che stabiliscano quali utenti possono accedere a determinati dati.

## **1.5 Funzionalità**

Oracle APEX fornisce tutti gli strumenti necessari volti a garantire risultati di successo per i decision making process aziendali. Di seguito verrà posta particolare attenzione su alcune funzioni che APEX mette a disposizione per risolvere le problematiche enterprise, quali il reporting, la conversione di qualsiasi applicazione web in Progressive Web App (PWA), l'interfaccia utente Universal Theme ed infine la gestione dei processi di autenticazione e autorizzazione.

### **1.5.1 Dati e Reporting**

Come già sottolineato in precedenza, la piattaforma APEX è specializzata non solo nella visualizzazione dei dati, ma anche nella modifica degli stessi; è proprio questa caratteristica a distinguerla da altri tool impiegati dai developer nel ramo della Business Intelligence. Tra i principali obiettivi enterprise, sicuramente troviamo la traduzione dei dati contenuti nei database in informazioni intellegibili ed utili ai processi decisionali che involgono gli end users.

Oracle APEX, con i suoi numerosi e differenti componenti di reporting dal design moderno, personalizzabile e di facile utilizzo, è in grado di soddisfare tutte le esigenze che riguardano la visualizzazione, la manipolazione, l'aggiornamento, l'elaborazione dei dati ed inoltre la conversione di questi ultimi in grafici.

Innanzitutto, un report è il risultato formattato di una query SQL, la quale può essere generata manualmente selezionando una tabella o una vista mediante una procedura guidata.

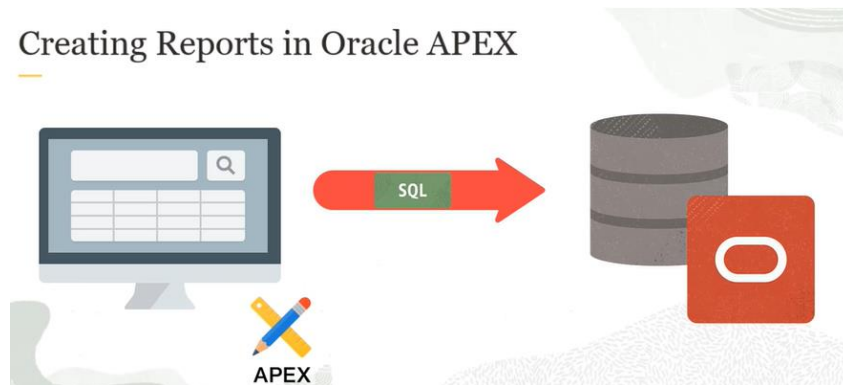


FIGURA 3: REPORT RISULTATO DI UNA QUERY SQL<sup>4</sup>

Tra le tipologie di report disponibili, i quali si differenziano per il livello di personalizzazione mediante il quale gli utenti intendono mostrare e gestire i dati (più precisamente ricercare, filtrare, ordinare, selezionare colonne d'interesse etc.), troviamo:

- **Classic Report:** è il componente di visualizzazione dei dati più semplice. I developer possono scegliere se selezionare una table da cui estrarre il report oppure utilizzare una query personalizzata. Offre funzionalità base per l'ordinamento delle colonne da mostrare e per il download dei dati.
- **Report con Form:** come sottolineato in precedenza, APEX differisce da altri strumenti di data visualization per la sua peculiare abilità nelle attività di modifica dei dati; queste ultime si concretizzano anche attraverso la creazione di pagine dell'applicazione contenenti report che includono form. Il risultato sarà, dunque, un report in cui per ogni record è presente un pulsante con icona che, se premuto, aprirà una form, in modal dialog o in modalità normale, il quale consentirà di apportare modifiche ai campi d'interesse.

<sup>4</sup> <https://apex.oracle.com/it/>

- **Interactive Report:** si distingue da un report classico principalmente per il più elevato grado di personalizzazione con riferimento alla visualizzazione e gestione dei dati. Quando un Interactive Report viene aggiunto in una page dell'applicazione, di default includerà una barra di ricerca, un menù "Actions", un menù di intestazione delle colonne, e un'icona per l'update di una riga. Mediante il menu azioni, gli utenti possono modificare il layout del report nascondendo colonne o selezionando quelle d'interesse. In aggiunta, si possono applicare filtri, definire ordinamenti, aggiungere funzioni di aggregazione dei dati e calcoli personalizzati, raggruppare record tramite un'istruzione di group by, creare grafici e molto altro ancora. Una volta terminata la customizzazione, il report può essere salvato e reso privato o pubblico.
- **Interactive Grid:** è un report che offre strumenti personalizzabili simili a quelli disponibili per l'Interactive Report ed inoltre, con un semplice drag and drop, è possibile trascinare le colonne, stabilendo l'ordine desiderato.
- **Faceted Search:** uno dei maggiori casi d'uso per cui viene utilizzato questo strumento di reporting concerne la realizzazione di siti e-commerce. All'interno di un report di tipo Faceted Search possono essere individuate due sezioni; l'area sinistra è riservata alla selezione dei valori di un determinato "facet" che si intende mostrare nel report, facilitando la ricerca, da parte degli utenti, dei dati d'interesse. Questa modalità di filtraggio avviene in real-time. Mentre la parte destra può apparire come un report classico o con cards.
- **Smart Filters:** è un componente APEX mostrato all'interno di una pagina dell'applicazione sotto forma di una barra di ricerca, all'interno della quale sono visibili i filtri applicati relativi ai campi del report sottostante. I filtri da applicare sono disponibili immediatamente sotto la barra di ricerca e vengono arricchiti con un'informazione inerente alla frequenza di un determinato valore all'interno del report.
- **Cards:** sono delle schede colorate che presentano le informazioni estratte da una table sottostante. Una card può includere icone, media, azioni, badges, URL, BLOB ed anche video.

### 1.5.2 Ambiente Mobile

In qualità di development tool non solo di applicazioni web, ma anche mobile, Oracle APEX dimostra ulteriormente la sua peculiare efficienza. Infatti, gli sviluppatori sono in grado di creare applicazioni web che possono essere facilmente installate in qualsiasi dispositivo mobile

da parte degli end users; le applicazioni in questione sono classificabili come Progressive Web Apps (PWA) le quali, viste come combinazione di web e mobile apps, agiscono e hanno sembianze di app native.

I vantaggi di tale funzionalità sono evidenti: il termine progressive rimanda alla fruibilità da parte di ogni utente indipendentemente dal browser scelto; si comportano come app native e si adattano a qualsiasi device di tipo desktop, smartphone e tablet; sono installabili sulla home del dispositivo utente e linkabili facilmente mediante condivisione dell'URL; i componenti APEX come reports, forms, grafici, calendari ed altro ancora, conservano le stesse proprietà e funzionalità in ambiente mobile. È, peraltro, utile sottolineare la semplicità di conversione ed installazione di una applicazione APEX in un device mobile, infatti, occorre semplicemente innestare un'impostazione all'interno dell'environment di sviluppo per fruire di tale funzione.

### **1.5.3 Universal Theme**

Ai fini di semplificare la user experience e la creazione di web application ad opera sia di novelli che di più esperti developer, APEX propone una interfaccia utente nota come Universal Theme. Tale user interface funge, infatti, da guida per l'utente per conoscere o accrescere le proprie competenze in merito alle funzionalità, ai tool ed i componenti offerti da Oracle Application Express e da ausilio per coloro che possiedono una limitata padronanza nell'uso di linguaggi quali HTML, CSS e JavaScript. Universal Theme può essere aggiunta al proprio ambiente di lavoro APEX e visibile assieme alle altre applicazioni già create ed inoltre può essere installata sui dispositivi desktop e mobile. Trattasi di una vera e propria applicazione divisa in molteplici sezioni, consultabili mediante un menu di navigazione, che includono informazioni inerenti al design di un'applicazione, a tutti i componenti APEX, alle web application icons ed altre utilities per tool HTML, CSS e JavaScript.

La sezione Design è ripartita in unità contenenti guide per l'utilizzo delle classi di colore CSS disponibili per i vari componenti User Interface (UI) di markup HTML dell'applicazione e per le tipologie di menu di navigazione che un developer può sfruttare; essa può, inoltre, fornire indicazioni sul template di una pagina web e una overview sui theme styles offerti da APEX. Un altro ramo del menu di navigazione consiste in un elenco dei componenti che la piattaforma propone, tra cui tutti i report citati in precedenza, il Calendario, i pulsanti, Badge List, Activity Timeline e molto altro. Selezionandone uno d'interesse, Universal Theme offre una serie di informazioni tra cui la descrizione, le istruzioni per l'inserimento di tale componente



nell'applicazione e una demo. A seguire, una sezione dedicata interamente alle icone che possono arricchire i componenti UI.

### **1.5.4 Autenticazione e Autorizzazione**

Durante lo sviluppo di un'applicazione APEX, il developer, a meno che non sia di carattere pubblico, vuole accertarsi che solo utenti autorizzati possano accedervi. Per garantire sicurezza all'interno della propria app, APEX fornisce i servizi di Autenticazione e Autorizzazione.

L'autenticazione viene realizzata attraverso una pagina di login che si accerta che un determinato user possieda le corrette credenziali per accedere all'applicazione; se l'username e password sono conformi, l'utente può definirsi autenticato.

Il meccanismo di autorizzazione invece consente di limitare i privilegi d'accesso di un utente a determinate pagine o componenti dell'applicazione.

L'autenticazione può essere inclusa selezionando un metodo di autenticazione già preconfigurato, creando uno schema di autenticazione personalizzato mediante funzioni PL/SQL eseguite prima di accedere ad una determinata pagina, oppure è possibile non utilizzare alcuno schema di autorizzazione rendendo l'app accessibile ad ogni gruppo di end users.

L'autorizzazione può essere implementata all'interno di un'applicazione APEX in riferimento a pagine o componenti specifici secondo due modalità:

1. creando uno schema di autorizzazione o copiandone uno esistente.
2. realizzando una pagina "Access Control Page", settando gli utenti autorizzati ed infine applicando questo schema ai componenti d'interesse dell'applicazione.

## **1.6 Oracle Application Express Development Environment**

Nel seguente paragrafo verranno introdotti e conseguentemente approfonditi i principali componenti dell'Integrated Development Environment (IDE) di APEX; verrà posta particolare attenzione ai moduli denominati Application Builder e SQL Workshop, mentre le sezioni Team Development e Gallery verranno solamente accennate.

Prima di focalizzare l'attenzione sugli elementi sopra elencati, è fondamentale introdurre il concetto di Workspace APEX.

Al fine di utilizzare i servizi dell'IDE di APEX per la creazione di applicazioni, è necessario dapprima creare un Workspace. Un Workspace è un'area di lavoro, un container logico che contiene tutte le risorse necessarie allo sviluppo e consente ad un delineato gruppo di utenti, con diversi gradi di accesso e funzionalità, di accedervi. APEX è un ambiente di lavoro strutturato in uno o più Workspaces, a ciascuno di essi è associato uno o più database schemas. Per cui, un particolare Workspace consente solo agli utenti abilitati di accedere ai dati, agli oggetti e alle applicazioni create all'interno di quell'area di lavoro. Una volta creata un'area di lavoro con ID e nome univoco, è possibile accedervi compilando una login form page con campi relativi al nome del workspace, l'username e password dell'utente.

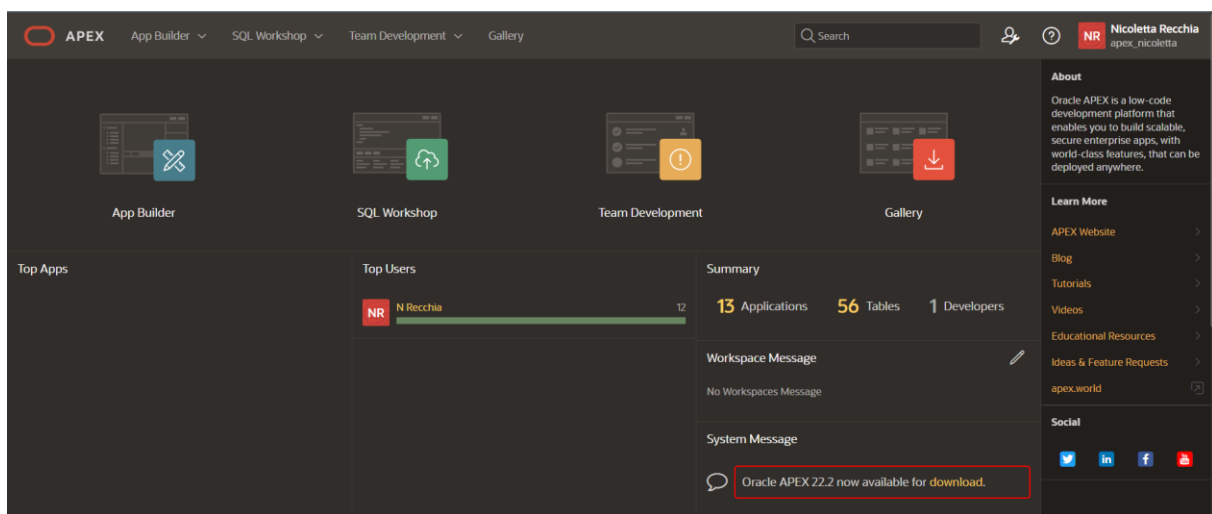
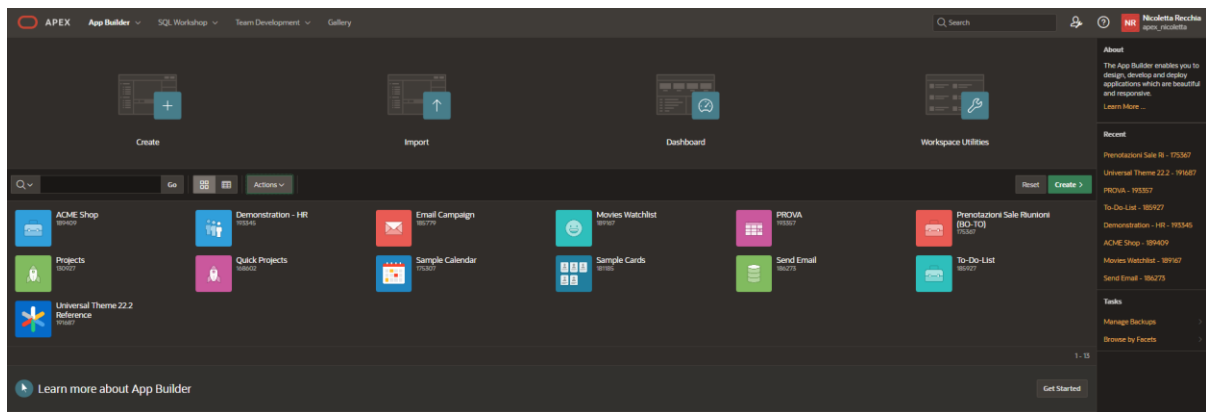


FIGURA 4: HOME PAGE WORKSPACE APEX

L'immagine sovrastante mostra l'IDE di APEX, nonché home page del Workspace a cui un utente viene indirizzato direttamente dalla pagina di login, con i suoi principali tool per la creazione e gestione delle applicazioni.

### 1.6.1 Application Builder

L'Application Builder rende possibile la progettazione, sviluppo e distribuzione di applicazioni di qualità. Le primarie attività che coinvolgono l'App Builder sono la creazione di nuove applicazioni, la modifica e l'import di una preesistente, la visualizzazione della Dashboard e l'accesso alle Workspace utilities.



**FIGURA 5: APPLICATION BUILDER**

La figura presenta l'Home Page dell'App Builder nella quale vengono visualizzate tutte le applicazioni create o installate in precedenza in quella specifica area di lavoro; essa si compone, inoltre, di quattro moduli:

- **Create:** indirizza direttamente alla procedura guidata Create Application Wizard per la creazione di nuove applicazioni
- **Import:** utilizzato per importare applicazioni precedentemente esportate oppure sviluppate altrove.
- **Dashboard:** fornisce informazioni come il numero di applicazioni e le relative pagine in quel determinato workspace, le pagine più attive ed altre metriche.
- **Workspace Utilities:** offre molteplici servizi, tra cui l'Export di applicazioni o altri elementi del workspace in un file script SQL.

## 1.6.2 SQL Workshop

Modulo dell'APEX Development Environment che consente di definire, mantenere e gestire tutti gli oggetti dell'infrastruttura di un database, tra cui tabelle, viste, indici, triggers etc., di visualizzarli nel browser ed inoltre, di eseguire queries SQL o PL/SQL.

L'immagine successiva mostra una panoramica di SQL Workshop.

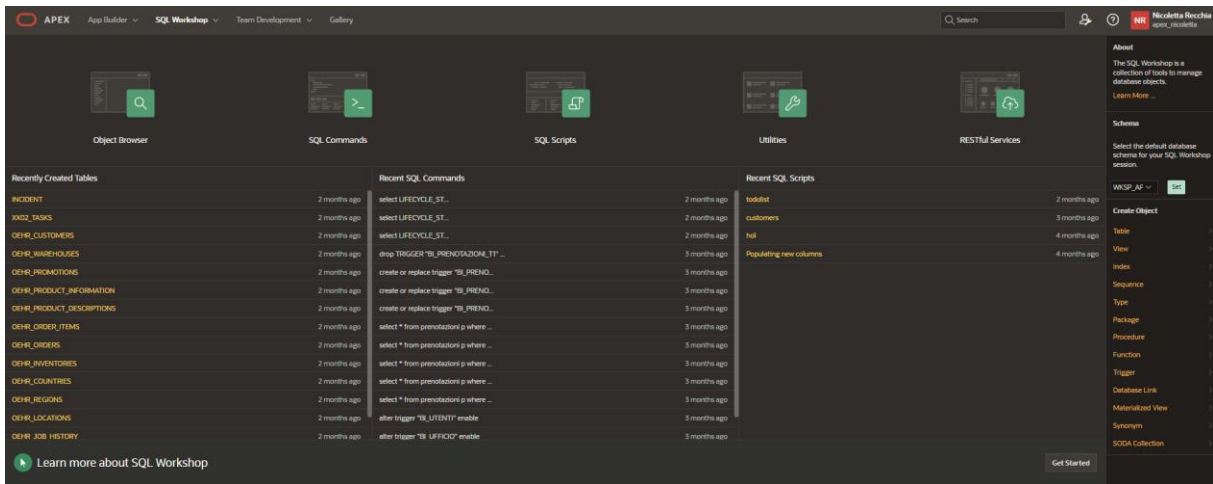


FIGURA 6: SQL WORKSHOP

I cinque moduli di SQL Workshop si compongono di:

- **Object Browser:** tool che si occupa della gestione degli oggetti del database.

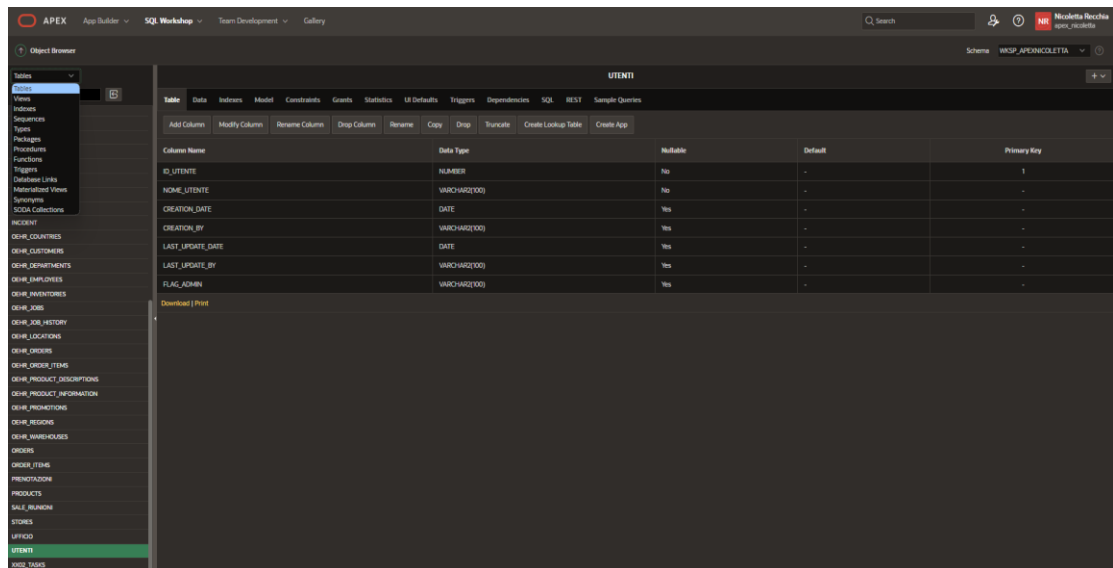


FIGURA 7: OBJECT BROWSER

Com'è visibile in figura, Object Browser è suddiviso in due sezioni; la sezione sinistra mostra l'elenco degli elementi appartenenti ad un oggetto selezionato, in questo caso le tabelle ed in particolare quella "UTENTI". La sezione destra presenta le informazioni relative all'oggetto scelto (nello specifico vengono mostrati tutti i campi della tabella "UTENTI" con i relativi datatype) e all'interno di questa sarà eventualmente possibile apportare modifiche sulle colonne; inoltre, è possibile visualizzare tutti i dati contenuti nella table (Data), gli indici (Indexes), i tipi di vincoli esistenti (Constraints), i Triggers

ed altro ancora. Infine, cliccando sul pulsante “+” è possibile creare qualsiasi tipo di oggetto.

- **SQL Commands:** componente utilizzato per la modifica, visualizzazione, inserimento ed esecuzione di oggetti del database mediante istruzioni SQL o blocchi PL/SQL. È anch'esso diviso in due parti: la parte superiore è rappresentata dall'Editor dei Comandi dedicato all'esecuzione delle query; la parte inferiore presenta i risultati ottenuti dall'esecuzione della query sovrastante, i quali sono disponibili per il download sotto forma di foglio di calcolo. L'immagine sottostante mostra la struttura di SQL Commands con un esempio di query sulla table “Ufficio” ed i relativi risultati.

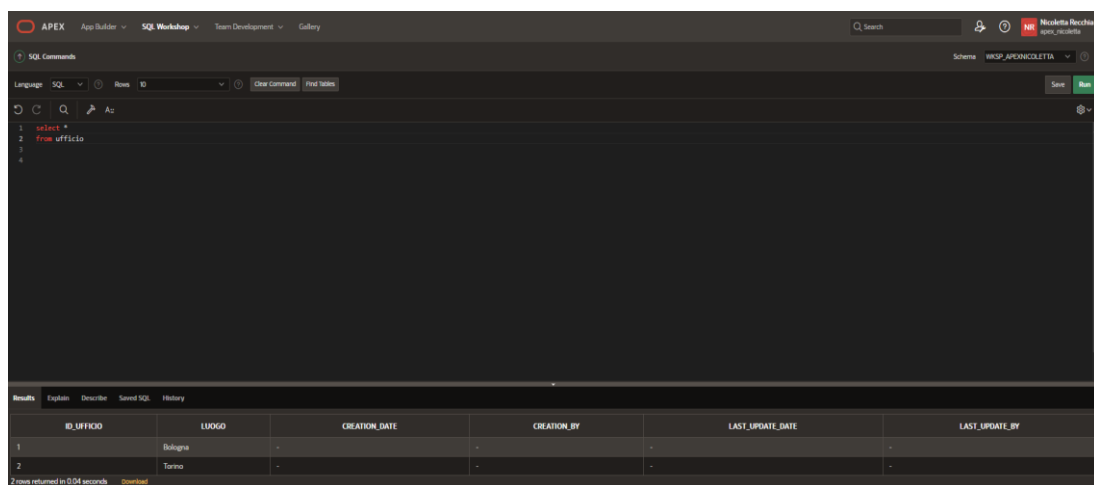


FIGURA 8: SQL COMMANDS

- **SQL Scripts:** modulo impiegato per la creazione e gestione di script SQL e PL/SQL. Uno script viene individuato come un insieme di istruzioni SQL o blocchi di istruzioni PL/SQL e può essere creato, modificato, visualizzato, eseguito e salvato direttamente nel repository di SQL Scripts; quest'ultimo consente, inoltre, di caricare o scaricare scripts all'interno del filesystem locale.
- **Utilities:** come è visibile nella figura 9, è possibile usufruire del componente Utilities per svariati servizi tra i quali, caricare o scaricare dati in forma di file CSV, XLSX, XML o JSON, ripristinare oggetti del database eliminati, installare datasets di esempio con Sample Datasets per costruire semplici applicazioni, monitorare il database, generare query con Query Builder, progettare e creare rapidamente prototipi di modelli di dati mediante sintassi abbreviata del linguaggio SQL tramite Quick SQL. Quest'ultima utility, infatti, rappresenta una delle maggiori innovazioni offerte dal tool APEX, è uno

strumento che genera automaticamente oggetti di un database avendo come input istruzioni con sintassi estremamente abbreviata, supportando il developer nell'apprendimento delle procedure di gestione dei dati. La figura 10, mostra il codice Quick SQL nella parte sinistra, mentre nella sezione destra le stesse istruzioni tradotte in query SQL.

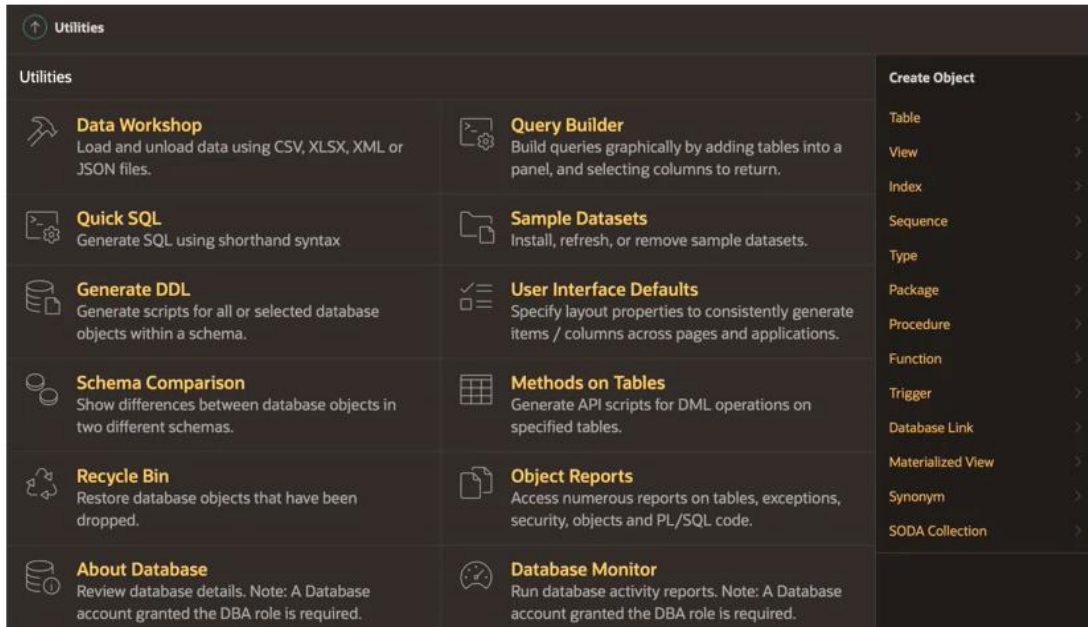


FIGURA 9: UTILITIES

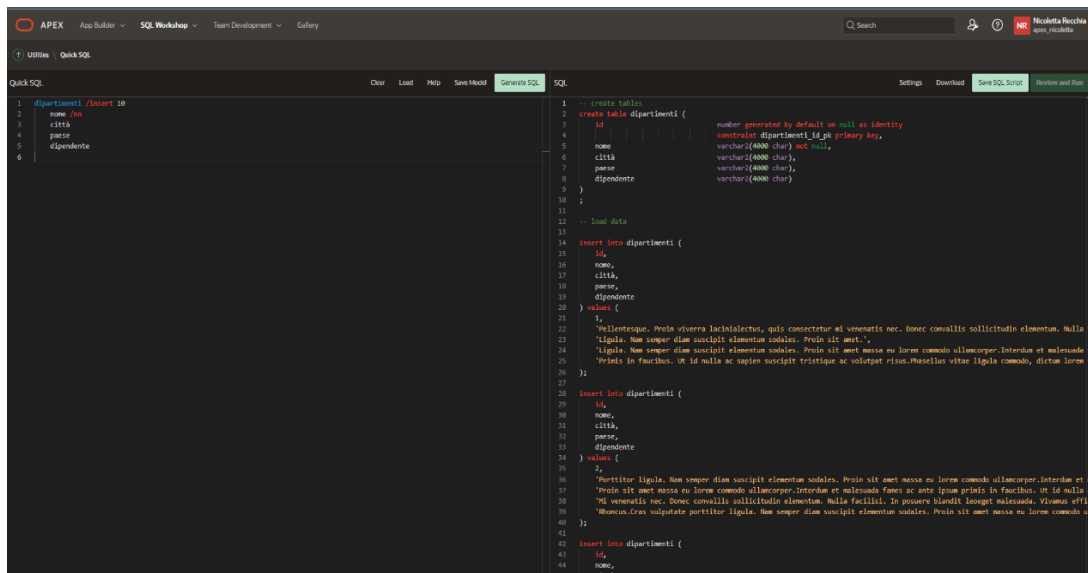


FIGURA 10: QUICK SQL

- **RESTful Services:** modulo utilizzato per definire Web Services usando query SQL o PL/SQL sul database.

### **1.6.3 Team Development e Gallery**

Team Development è uno strumento APEX ideato per mantenere traccia e gestire problemi mediante la creazione di template allo scopo di fornire ai developer ed agli utenti statistiche relative a processi, bugs, feedback.

Gallery rende disponibili esempi di applicazioni presenti nel repository di Oracle APEX che possono essere installate in un workspace, fungendo da guida nella creazione di applicazioni soprattutto agli junior developers APEX.

## 1.7 Anatomia di un'applicazione Oracle APEX

Come delineato nel paragrafo precedente, uno sviluppatore può creare multipli workspaces.

Un workspace può essere visto come una repository virtuale in cui risiedono le applicazioni create. Un'applicazione APEX si compone di due o più pagine web: ciascuna pagina web può contenere diverse aree, chiamate Regions, necessari a mostrare e ricevere informazioni e può includere Dynamic Actions e Page Processes, utili, rispettivamente, alla generazione di azioni quando si verifica una determinata condizione e all'esecuzione di un'azione durante il rendering o la submission di una pagina; una Region può contenere Page Items e Buttons.

Di seguito è raffigurata la struttura di un'applicazione APEX a partire dal workspace. <sup>5</sup>

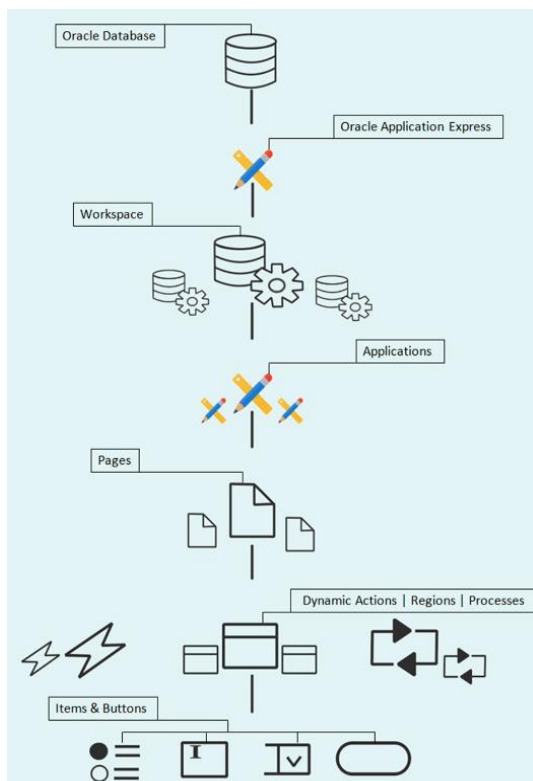


FIGURA 11: ANATOMIA DI UN'APPLICAZIONE APEX

<sup>5</sup> Oracle APEX 20 For Beginners: a platform to develop stunning, scalable data-centric web apps fast, Riaz Ahmed



## 1.7.1 Componenti di un'applicazione

Un'applicazione in Oracle APEX è una collezione di pagine connesse tra loro mediante un menu di navigazione, schede, pulsanti o hypertext links. Una pagina è l'elemento costitutivo di un'applicazione e può contenere diversi elementi dell'interfaccia utente quali pulsanti, page items, menu di navigazione ed altro, all'interno di aree chiamate regions; contiene la logica dell'applicazione mostrando elementi quali report, grafici, calendari, forms e tutti gli altri componenti APEX disponibili, svolgendo inoltre calcoli e convalide. APEX esegue il rendering delle pagine dinamicamente ed elabora le pagine basandosi sui dati presenti nel database. Per visualizzare una pagina, APEX esegue due processi per il rendering e il processamento delle pagine; il processo Show Page dedicato al rendering, unisce tutti i componenti di una pagina e li traduce in una pagina HTML visualizzabile; il processo Accept Page per il processamento di una pagina, si occupa di effettuare calcoli, validazioni, processi e branching.

Una Pagina è identificata da un numero identificativo univoco e può essere creata mediante una procedura guidata, oppure si può creare una pagina vuota e successivamente aggiungere i componenti d'interesse. Ad una region può essere associato un determinato template, in modo tale da stabilirne l'aspetto, la dimensione all'interno della pagina, la posizione dei vari componenti di reportistica, items, pulsanti e menu di navigazione. Un page item può rappresentare un campo o un'area di testo, una lista di selezione ed altro ancora; esso viene comunemente riconosciuto all'interno dell'APEX environment con una sintassi ben precisa: una "P" posta davanti al numero della pagina e a seguire un nome descrittivo che rimanda al contenuto dell'item. Ad esempio "P8\_NOMEDIPARTIMENTO", indica un page item di pagina 8, che conterrà informazioni inerenti al nome di un dipartimento. Un pulsante o button, viene utilizzato per indirizzare l'utente ad un'altra pagina, ad un URL, generare, elaborare o apportare modifiche ai dati mediante submission della pagina. Infine, all'interno di una region possono essere inclusi controlli di navigazione quali l'elenco menu di navigazione, la barra di navigazione e percorsi (Breadcrumbs).

## 1.7.2 Page Designer

Come enunciato nei precedenti paragrafi, una volta acceduti in un determinato workspace, il componente APEX che consente la creazione, la gestione e la modifica delle applicazioni è l'Application Builder. Quest'ultimo, presenta un catalogo delle applicazioni già create all'interno dell'area di lavoro. Lo strumento dell'ambiente di sviluppo APEX mediante il quale una nuova pagina di un'applicazione viene gestita, oppure ne viene migliorata una preesistente, è il Page Designer.

Il Page Designer rappresenta un ambiente di sviluppo integrato completo di tutte le funzionalità utili alla gestione degli elementi presenti in una pagina; è l'interfaccia di sviluppo integrata in APEX utilizzata per la visualizzazione, la creazione e la modifica dei componenti che racchiudono la logica dell'applicazione.

L'immagine 12 mostra il Page Designer Interface inerente a una pagina di tipo form di un'applicazione esistente, a cui è possibile accedere dall'App Builder selezionando l'applicazione e successivamente la pagina d'interesse.

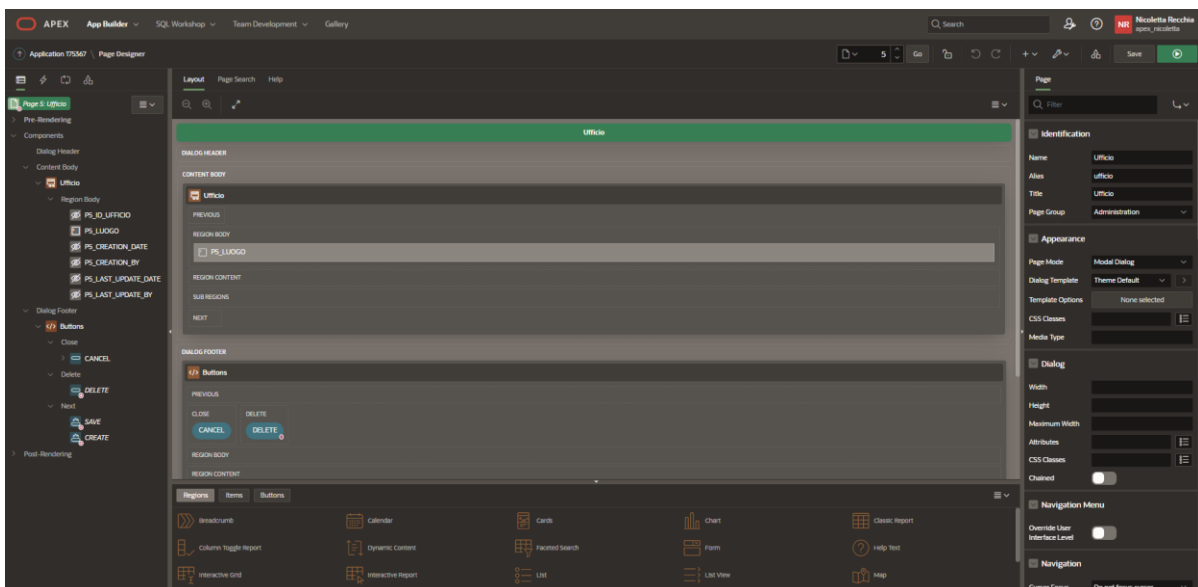


FIGURA 12: PAGE DESIGNER

Il Page Designer è suddiviso in diverse sezioni:

- **Toolbar:** la toolbar, o barra degli strumenti, incorpora diverse funzioni, tra cui la ricerca e il blocco/sblocco di una pagina per limitare solo ai developer autorizzati di apportare

modifiche, l'annullamento o il ripristino di una modifica effettuata, il salvataggio, l'esecuzione e l'eliminazione della pagina corrente ed altro ancora.



FIGURA 13: PAGE DESIGNER TOOLBAR

- **Pannello sinistro o Tree pane:** suddiviso anch'esso in più schede; in particolare, una per il Rendering, la sezione per le Dynamic Actions ed infine quelle per il Processing e per gli Shared Components, tutte raffigurate con una struttura ad albero. Esso prevede, altresì, di mostrare gli elementi di una pagina, di gestire i processi, i calcoli e le convalide relative alla logica applicativa ed inoltre gestisce Dynamic Actions e Shared Components. Di seguito verranno approfondite le diverse aree, illustrando per ciascuna di essa una immagine esplicativa.

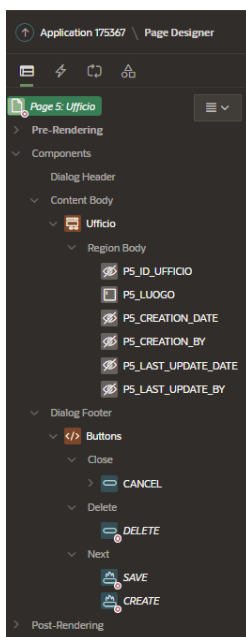


FIGURA 14: SCHEDA DI RENDERING NEL TREE PANE

La scheda di Rendering contiene tutte le informazioni ovvero le regioni, i page items, i pulsanti presenti nella pagina. Com'è possibile vedere in figura, la pagina presenta un'unica regione chiamata "Ufficio", la quale contiene nel suo Body diversi page items ("P5\_ID\_UFFICIO", "P5\_LUOGO", "P5\_CREATION\_DATE", "P5\_CREATION BY", "P5\_LAST\_UPDATE\_BY") che rappresentano dati contenuti nello schema a cui il workspace dell'applicazione punta; nella sezione Dialog Footer della region, sono presenti diversi pulsanti quali, "CANCEL", "DELETE", "SAVE" e "CREATE". Nello

stesso vi sono due rami della struttura ad albero, uno dedicato al Pre-Rendering per il processamento di calcoli eseguiti prima del rendering della pagina ed un altro per il Post-Rendering che prevede elaborazioni dopo il rendering di una pagina.

La scheda relativa alle Dynamic Actions supporta la creazione o visualizzazione delle azioni inerenti ai comportamenti client-side, senza necessità di ricorrere al coding in JavaScript.

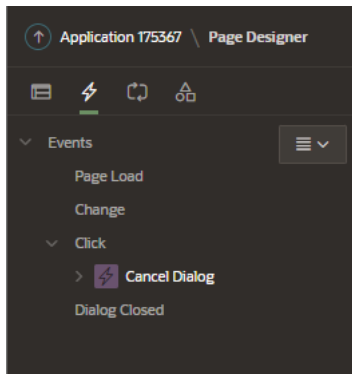


FIGURA 15: SCHEDA DYNAMIC ACTIONS NEL TREE PANE

La sezione per il Processing è dedicata alla definizione delle logiche dell'applicazione come calcoli, convalide e processi. Si possono esplicitare calcoli necessari a settare i valori dei page items nonché processi per la manipolazione dei dati, convalide per verificare se i requisiti di accuratezza, consistenza e qualità siano rispettati.

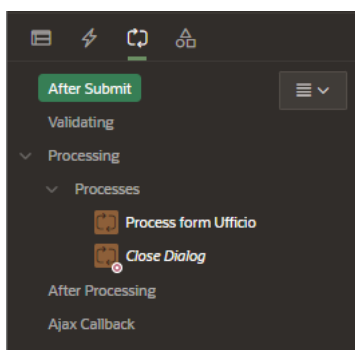
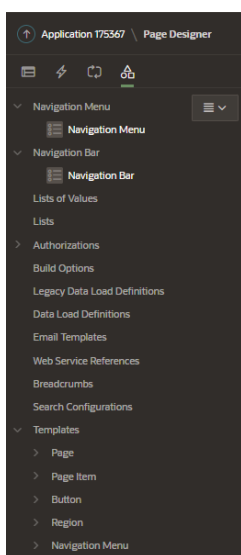


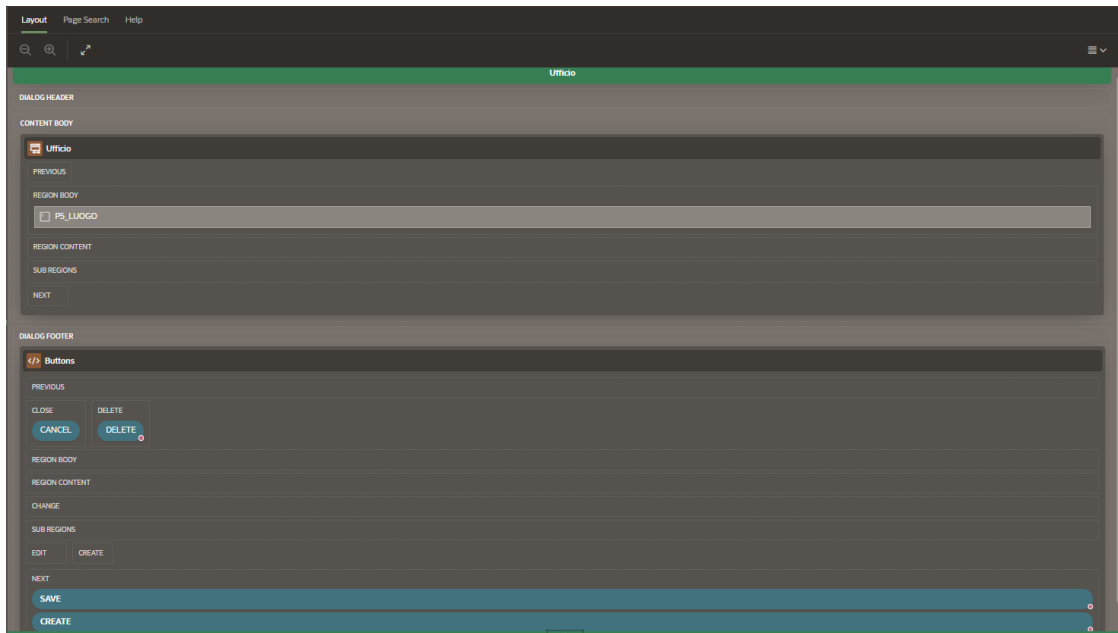
FIGURA 16: SCHEDA PER IL PROCESSING NEL TREE PANE

Infine, l'ultima scheda è quella riservata alle informazioni relative agli Shared Components; essi sono degli elementi che, una volta definiti, possono essere applicati e mostrati in qualsiasi pagina dell'applicazione.



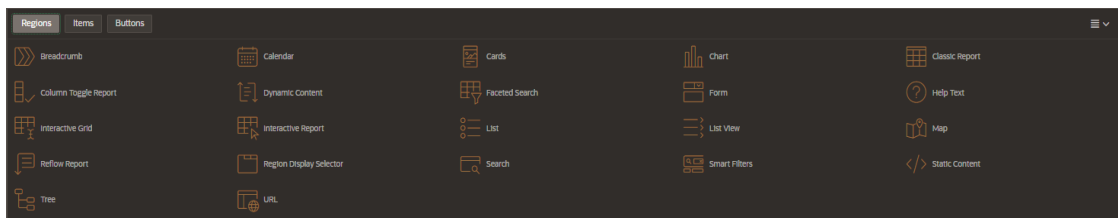
**FIGURA 17: SCHEDA SHARED COMPONENTS NEL TREE PANE**

- **Pannello Centrale:** è diviso in due sezioni, di cui la parte superiore contiene tre schede navigabili, ossia, Layout, Page Search e Help. La scheda di Layout è una rappresentazione grafica degli elementi presenti, un'anteprima di ciò che verrà mostrato nella pagina, ovvero, le regions, page items e bottoni. Ciascuno degli elementi appena citati può essere aggiunto, anche con una semplice azione di drag and drop, dalla Gallery sottostante; Page Search è una utility che facilita la ricerca di tutti i metadati presenti nella pagina; la sottosezione Help è utile per descrivere in forma testuale una determinata proprietà selezionata nel pannello di destra.

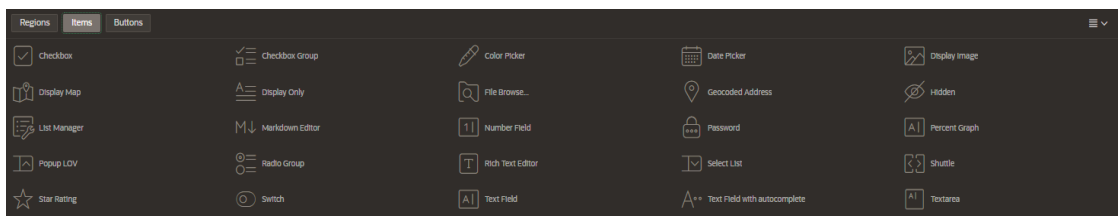


**FIGURA 18: PANNELLO CENTRALE**

La parte inferiore del pannello centrale è riservata alla Gallery, la quale offre una lista di tutti i tipi di regions, items e pulsanti che APEX mette a disposizione per essere aggiunti alla pagina.



**FIGURA 19: SCHEDA REGION NELLA GALLERY**



**FIGURA 20: SCHEDA ITEMS NELLA GALLERY**

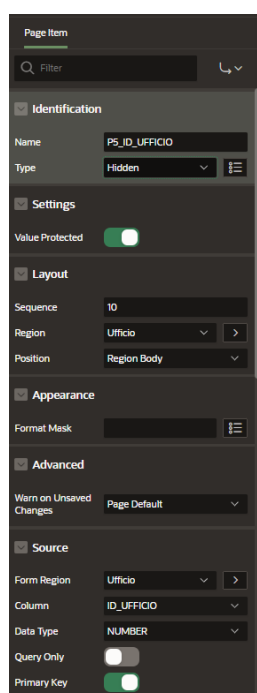


**FIGURA 21: SCHEDA BUTTONS NELLA GALLERY**

- Pannello di destra o Property Editor:** mostra tutte le proprietà e i valori associati ad un componente della pagina; infatti, selezionando un elemento, che sia una region, un page item o button (direttamente dal pannello sinistro o dal Layout centrale) nel Property Editor vengono mostrate tutte le specifiche relative all'Identificazione (Nome, Titolo, Alias del componente), all'Appearance ( il template utilizzato per mostrare quell'elemento), alla Source (ad indicare la sorgente del dato a cui è associato l'elemento, se prelevato direttamente da una tabella o risultato di una query) ed altro ancora.

Ad esempio, selezionando nella scheda di Rendering il Page Item “P5\_ID\_UFFICIO”, il Property Editor mostrerà tutte le informazioni relative ad esso.

Alcune di esse, come visibili della figura mostrata di seguito, indicano che è un item di tipo “Hidden” ossia che non verrà mostrato nella pagina all'esecuzione. Esso si trova nel body della region a cui appartiene e che ha come sorgente dati la colonna denominata “ID\_UFFICIO” con datatype NUMBER.



**FIGURA 22: PROPERTY EDITOR**

## 1.8 Creare un'applicazione APEX

Di seguito verranno brevemente descritte le procedure volte alla creazione di una nuova applicazione e alla generazione di una pagina.

Una volta che l'utente accede alla home page di un determinato Workspace, per la realizzazione di una nuova applicazione, è necessario entrare nel modulo App Builder. È possibile visionare le applicazioni già create in precedenza, modificarle, importare altre applicazioni, crearne di nuove o installarle dalla Gallery, oppure rimuoverle dal workspace.

Per creare una nuova applicazione, occorre selezionare il pulsante "Create" che indirizza ad una pagina "Create an Application" che mostra tre modalità di realizzazione: la creazione di una nuova app basata su tabelle di dati esistenti o fornendo un'istruzione SQL; il caricamento di dati da un file, al fine di creare un'applicazione su quei dati; la terza opzione riguarda, invece, l'installazione di una starter o sample app. La figura sottostante mostra quanto appena detto.

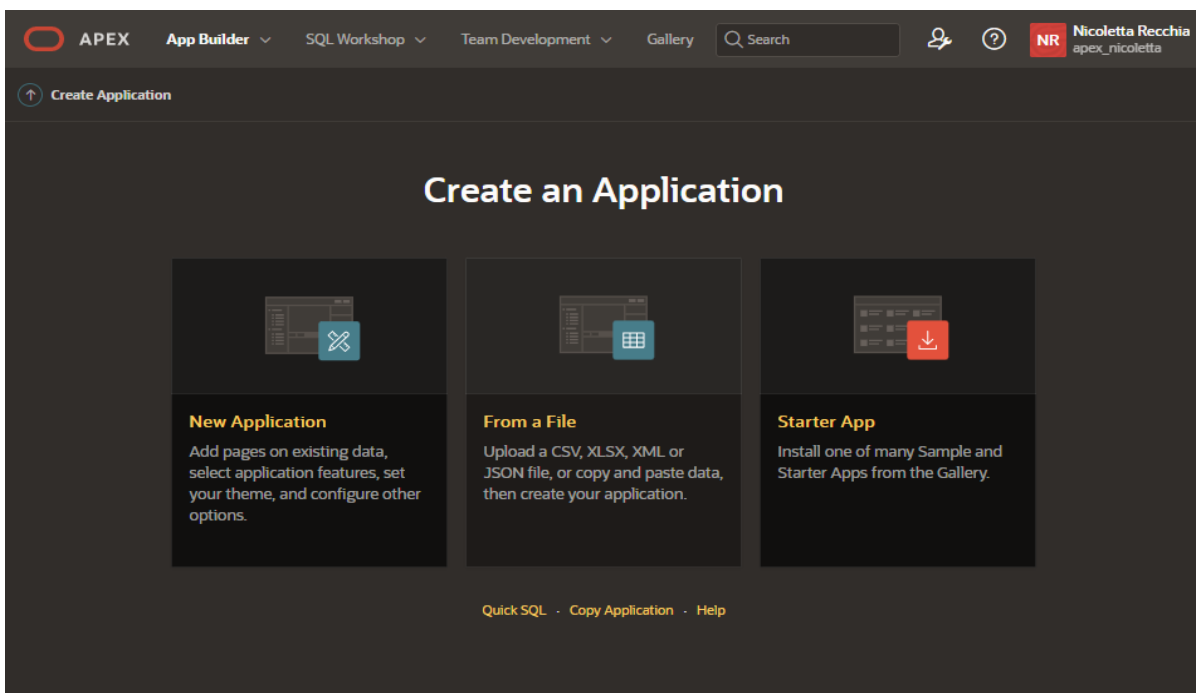
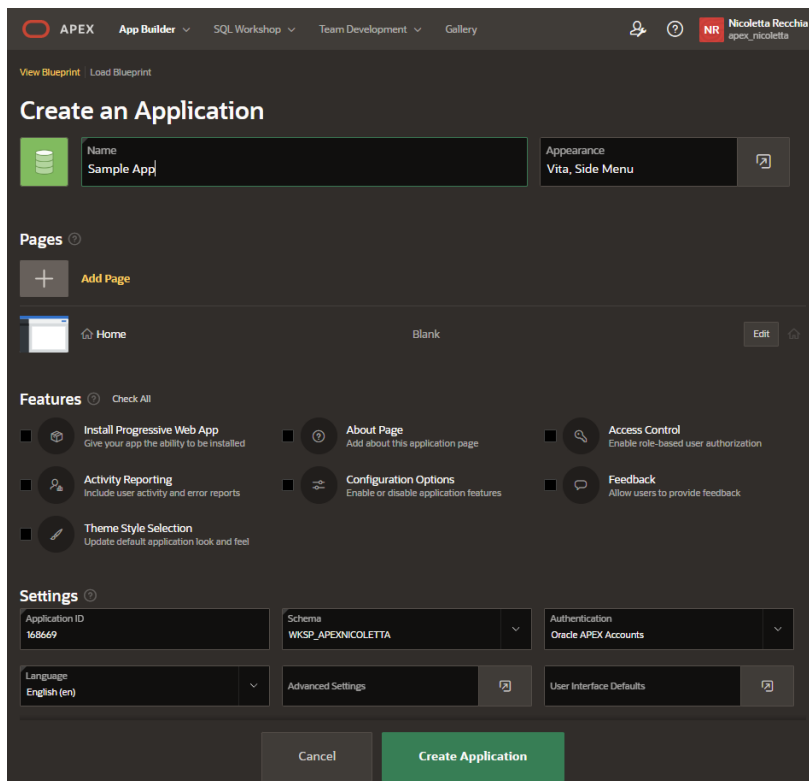


FIGURA 23: CREATE APPLICATION

Selezionando "New Application", si aprirà una nuova pagina in cui viene richiesto di inserire un nome, selezionare un'icona e l'aspetto desiderato per la pagina, più precisamente il Theme Style ed il tipo di menù di navigazione, selezionare alcune o tutte le Features elencate, come ad esempio "Install Progressive Web App" che consente all'app di essere installata su dispositivo



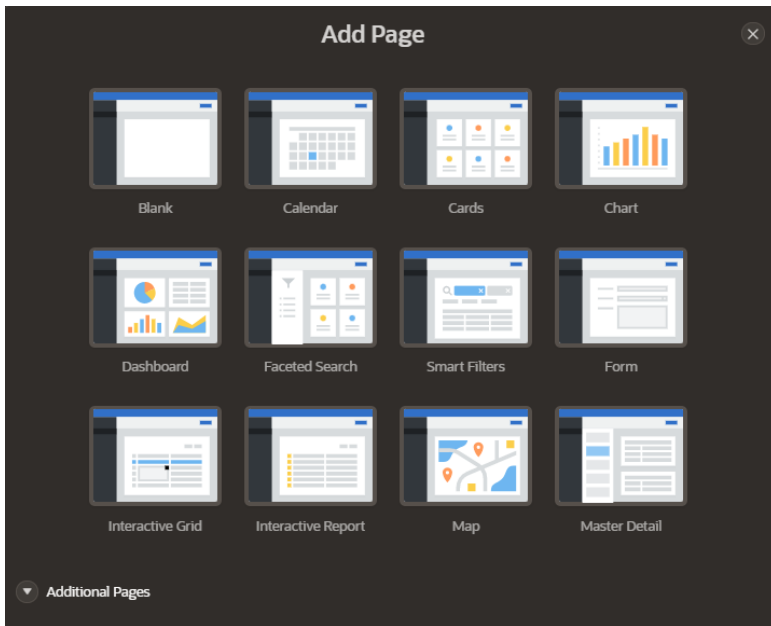
mobile, ed in fondo mostrerà alcune impostazioni tra cui l'ID univoco che verrà attribuito all'app e lo schema puntato dal Workspace.



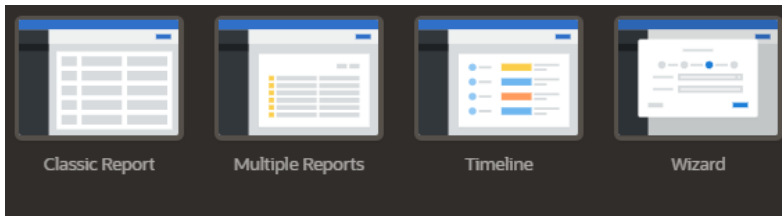
**FIGURA 24: CREATE AN APPLICATION**

Dalla figura 24 è visibile la sezione “Pages”, la quale è riservata alla creazione di nuove pagine. Si può inoltre notare che APEX crea automaticamente la home page dell'app, una pagina vuota, che può essere modificata o rimossa.

Cliccando su “Add Page”, l'utente viene reindirizzato ad un modulo che consente la creazione di una pagina mediante procedura guidata; è possibile, infatti, generare una pagina con il componente d'interesse, come un Classic Report, un Calendario, una Dashboard e molto altro ancora. La figura sottostante mostra appunto l'Add Page wizard.

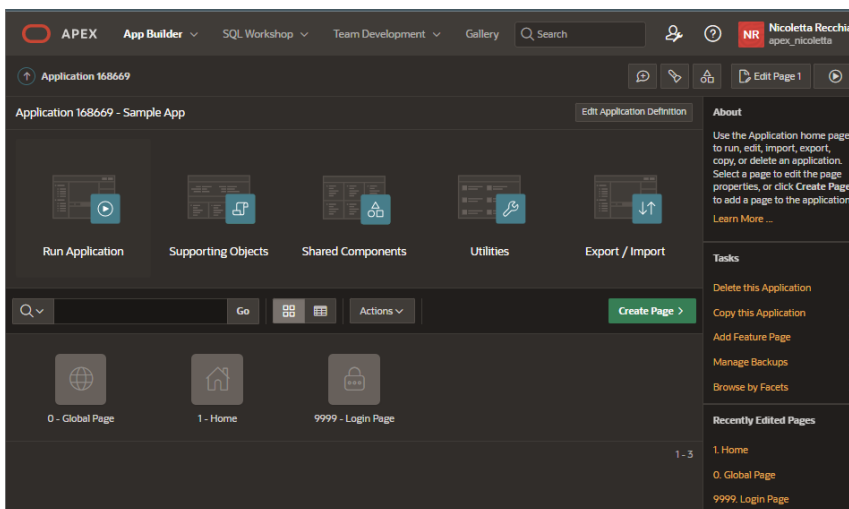


**FIGURA 25: ADD PAGE WIZARD**



**FIGURA 26: ADD PAGE WIZARD**

Una volta aggiunte tutte le pages mediante procedure guidate, il “Create Application” button (visibile nell’immagine n. 24) genera l’applicazione e rimanda all’Application home page che mostrerà l’elenco di tutte le pagine e una serie di possibili funzionalità, tra cui l’esecuzione, la modifica, l’import/export e la cancellazione.



**FIGURA 27: APPLICATION HOME PAGE**

La creazione di pagine (Create Page) può avvenire mediante procedura guidata selezionando una pagina che include già una region di una determinata tipologia, oppure selezionando “Blank Page”, il developer sarà libero di generarne una vuota, svilupparla interamente e personalizzarla mediante il Page Designer.

The screenshot shows a 'Create Blank Page' dialog box with the following configuration:

- Page Definition:**
  - Page Number: 2
  - Name: Students Report
  - Page Mode: Normal (selected), Modal Dialog, Drawer
- Navigation:**
  - Use Breadcrumb:
  - Breadcrumb Parent Entry: - No parent entry -
  - Use Navigation:
  - Navigation Preference: Create a new entry
  - Parent Navigation Menu Entry: - No parent selected -
  - Icon: fa-file-o

Buttons at the bottom: < Cancel Create Page

FIGURA 28: CREATE BLANK PAGE

Nell’immagine raffigurata sopra, viene creata una pagina denominata “Students Report” che può essere presentata normalmente (a schermo intero), sotto forma di Modal Dialog o Drawer. Quest’ultimo, nello specifico, rappresenta un particolare template per una finestra Modal Dialog. Lo sviluppatore può scegliere, inoltre, se mostrare il percorso della pagina (Use Breadcrumbs) ed inserirla nel menu di navigazione (Use Navigation). Una volta confermata la creazione, l’utente potrà iniziare a lavorare sulla pagina nel Page Designer, come mostrato nell’immagine di seguito.

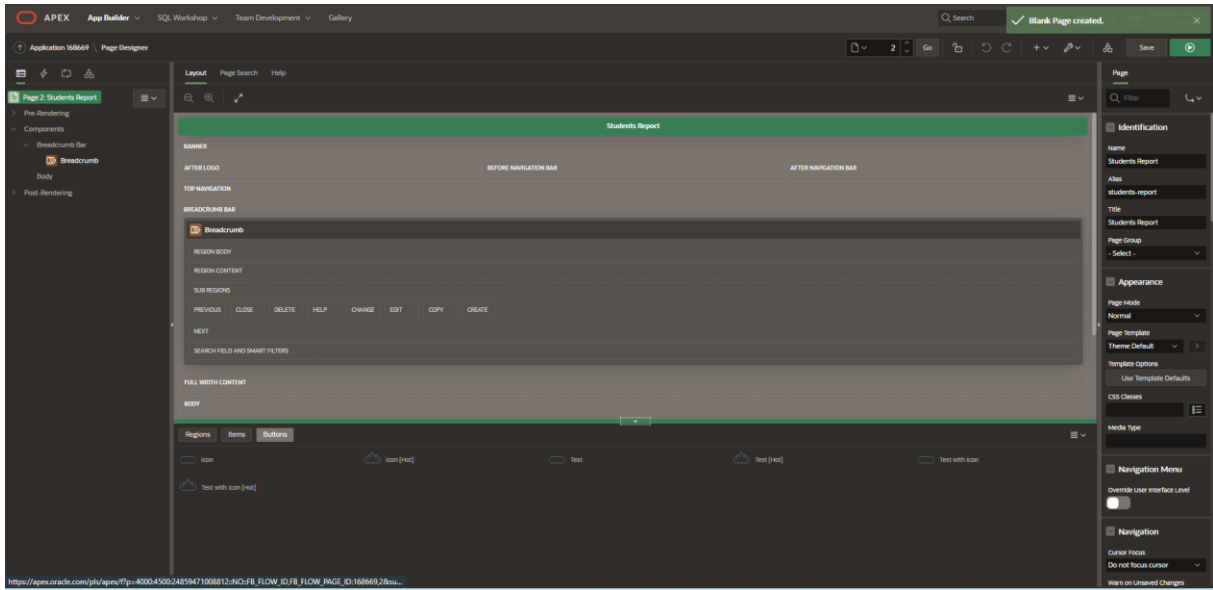


FIGURA 29: PAGE DESIGNER DI "STUDENTS REPORT"

# Capitolo 2

## AMS: Real-Time Incidents

In questo capitolo verranno esplicitati i requisiti discussi in fase di analisi e progettazione dell'applicazione web AMS:Real-Time Incidents per il monitoraggio real-time di sistemi business-critical, mediante piattaforma di sviluppo enterprise low code. Ne verranno descritte le funzionalità e gli obiettivi desiderati, ponendo particolare attenzione ai motivi che hanno favorito e reso necessario l'utilizzo del development tool utilizzato. Successivamente, verrà indagata con minuziosità la struttura di AMS: Real Time-Incidents, presentandone le logiche dietro ciascuna sezione dell'applicazione, le tecniche ed i componenti offerti dal framework low code per l'implementazione.

### 2.1 Analisi dei requisiti

L'applicazione AMS:Real-Time Incidents, che verrà utilizzata da una business unit dell'azienda Mediamente Consulting, sede di tirocinio, è strumentale all'attività di anomaly detection in real-time sui sistemi dei clienti aziendali sottoposti a costante monitoraggio. Il sistema sarà, pertanto, in grado di ricevere gli incidents, ossia eventi quali alert, errori ed eventuali irregolarità provenienti dai server dei clienti, classificati per livello di criticità. Al fine di garantire una migliore gestione degli incidents, l'applicazione consente la visualizzazione grafica di informazioni relative alla natura/tipologia, al grado di severità, nonché alla categoria di appartenenza, al sistema del cliente da cui originano, e la descrizione del problema riscontrato.

AMS: Real-Time Incidents è costituita da diverse pagine web, ognuna intenta a mostrare, mediante differenti strumenti di reportistica di dati, gli eventi rilevati indirizzandoli in diverse sezioni in base all'etichetta a cui appartengono, ciascuna associata ad un livello di pericolosità/complessità e ad una categoria, anch'essa distinta per grado di criticità. Gli eventi di cui si parla riguardano, ad esempio, la mancata disponibilità di servizi presenti sui server dei clienti, problemi inerenti alle percentuali di utilizzo della CPU e della memoria di un qualche componente dei sistemi clienti, oppure all'assenza di avvenuto backup dei database dei clienti. Si dovrà consentire all'utente di porre in evidenza gli incidents o nasconderli/rimuoverli, oppure

mostrarli tutti senza filtrare per categoria e criticità; un'area dell'applicazione sarà dedicata ad informazioni relative ai clienti, quali principalmente i contatti e dati concernenti all'infrastruttura dei sistemi monitorati. Inoltre, una sezione chiamata Cloud Control, indica se il sistema che rileva gli incident, denominato Enterprise Manager, sta funzionando oppure no.

Tutti i requisiti sopra elencati sono mostrati attraverso moderni ed interattivi tool di reportistica di dati; le pagine dell'applicazione sono navigabili attraverso un menù di navigazione o accessibili mediante collegamenti tra loro.

L'implementazione di tali requisiti è stata realizzata mediante la più diffusa piattaforma low code per il developing di enterprise web apps, Oracle Application Express, di cui si è a lungo discusso nel precedente capitolo. La decisione di adottare APEX come strumento di sviluppo del sistema di monitoraggio AMS:Real-Time Incidents deriva dalle innumerevoli funzionalità offerte che rispondono perfettamente ai requisiti delineati in fase di analisi; tra questi è di rilevante importanza sottolineare l'efficienza di APEX nella la realizzazione del prodotto finale in breve tempo, essendo un Rapid Application Development low coding tool, ma anche per le numerose componenti per la visualizzazione e manipolazione dei dati disponibili (report con differenti template, strumenti di filtraggio dati, forms, finestre modal dialog e percorsi di navigazione). L'utilizzo di tale piattaforma prevedeva la conoscenza del linguaggio di interrogazione di database SQL, PL/SQL ed anche dei linguaggi di markup HTML, CSS e JavaScript.

L'applicazione di anomaly detection implementata, include una pagina di login in cui l'utente accede mediante le proprie credenziali. Una volta acceduti, la prima pagina mostrata sarà l'home page denominata "Panoramica": questa pagina è divisa in diverse aree di cui alcune raggruppano gli incident per etichetta, categoria d'errore e livello di severità, offrendo, inoltre, la possibile filtrare gli incident per cliente; una sezione consente, invece, di visualizzare gli eventi messi in evidenza dagli end users, corrispondenti agli incidents presenti nelle aree appena citate, soggette al filtraggio per etichetta, categoria e criticità; un'altra sezione dell'app è riservata alla visualizzazione di tutti gli ultimi incidents rilevati senza vincoli di categoria o altre proprietà.

Mediante un menù di navigazione è possibile accedere ad altre pagine di cui si compone l'applicazione: una pagina "Incident" che presenta un Classic Report in cui ciascuna riga corrisponde ad un evento con diversi campi quali: l'identificativo, il messaggio descrittivo, il cliente, la categoria, il target ed il tipo di target; un'altra pagina, riportata nel menu di

navigazione, è “Cloud Control”, gestita tramite un report in cui ciascuna riga corrisponde ad un cliente con informazioni relative allo stato, “Up” o “Down”, del sistema di monitoraggio, ed inerenti all’orario dell’ultimo incident ricevuto per un cliente, all’ultima sincronizzazione e all’ultimo controllo. Il menu riporta anche la pagina “Clienti” che mostra un elenco di tutti i clienti, i cui sistemi vengono monitorati, e cliccando su uno di essi, l’utente ha la possibilità di conoscere informazioni generali utili alla gestione dell’incident, ai contatti del cliente ed altro ancora. Infine, il menu si conclude con le pagine “Bacheca”, “Centralino” e “Calendario”, non ancora implementate, e di cui si avrà maggiore riguardo nel terzo capitolo inerente concernente gli sviluppi futuri.

L’anatomia dell’applicazione e le logiche implementative dietro a ciascuna delle pagine web, verranno meglio approfondite nei prossimi paragrafi; verrà descritto precisamente quali componenti APEX abitano le regions di ciascuna pagina, le queries che consentono la visualizzazione e gestione dei dati, i processi eseguiti ed ulteriori tecniche implementate.

Pare opportuno sottolineare che le successive figure che verranno mostrate, in quanto contenenti informazioni riservate sono state opportunamente oscurate.

## **2.2 Home Page Panoramica**

Una volta che l’utente avrà effettuato l’accesso nel workspace che include l’applicazione AMS:Real-Time Incidents, egli potrà eseguirla ed immediatamente dopo gli verrà richiesto di inserire le proprie credenziali. Se l’autenticazione va a buon fine, la prima pagina renderizzata che l’end user visualizzerà sarà l’home page “Panoramica”. Questa pagina, come anticipato nel precedente paragrafo, può ritenersi suddivisa in due macro-sezioni, sinistra e destra, ciascuna ripartita in ulteriori aree.

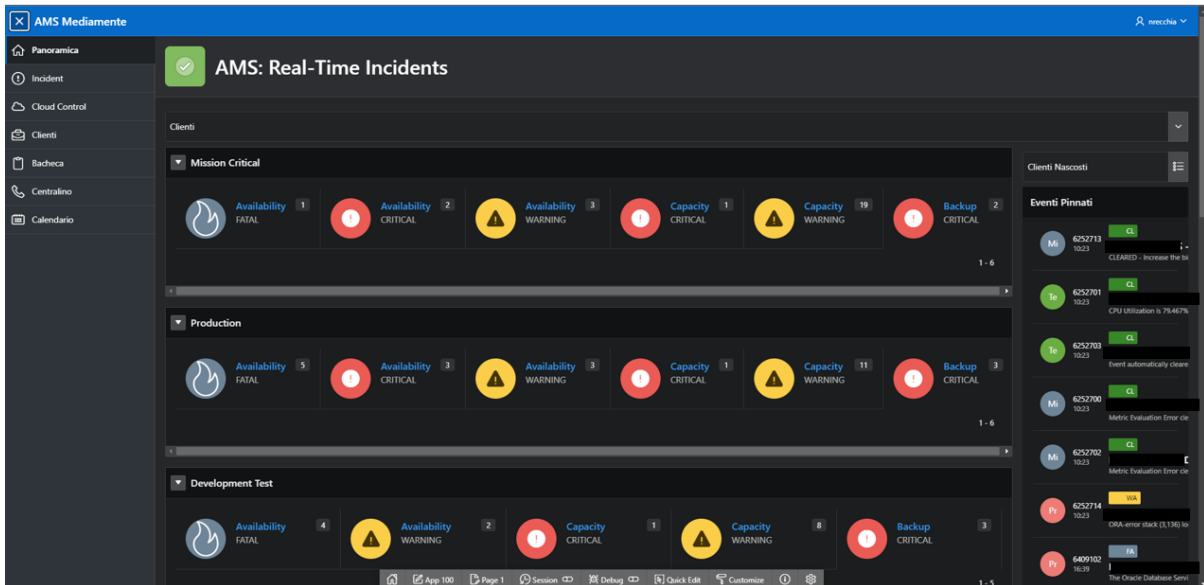


FIGURA 30: HOME PAGE PANORAMICA

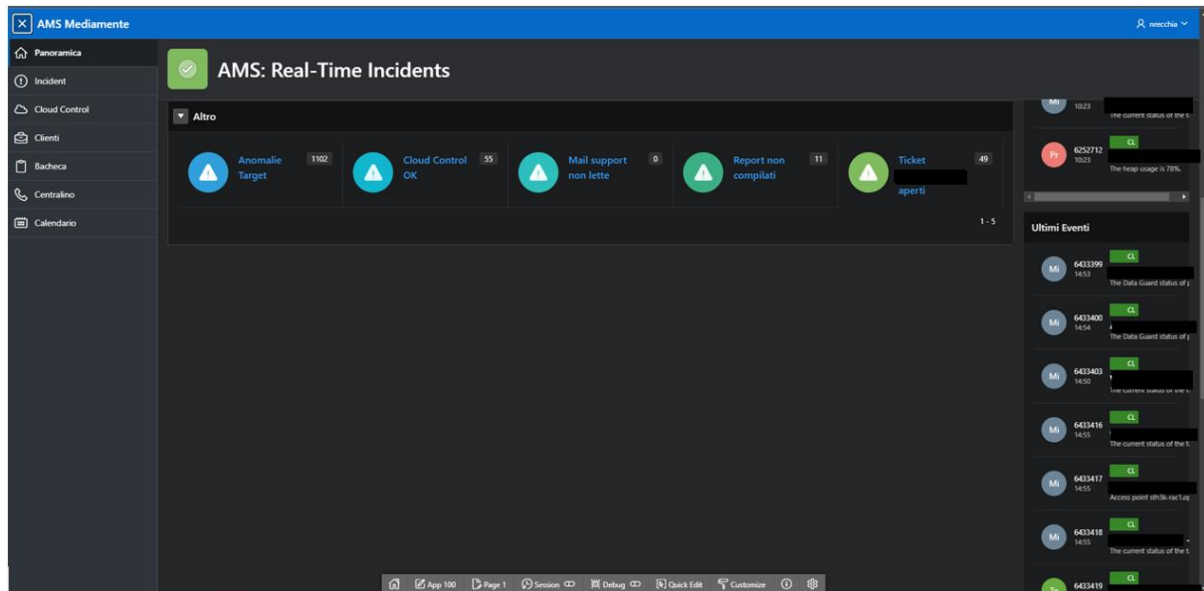


FIGURA 31: HOME PAGE PANORAMICA

Com'è possibile notare nelle immagini raffiguranti la pagina "Panoramica" di AMS:Real-Time Incidents, la sezione di sinistra è suddivisa in cinque aree principali: una barra di ricerca che avrà una funzione legata ai "Clienti" e le sezioni denominate "Mission Critical", "Production", "Development Test" ed infine "Altro". Le suddette sezioni, com'è intuibile, vengono presentate con lo stesso template del componente APEX utilizzato per visualizzare i dati mostrati; ciascuna delle prime tre aree "Mission Critical", "Production" e "Development Test" appena citate, si riferiscono, rispettivamente, ad una etichetta, distinta per il livello di importanza dei sistemi target dei clienti su cui sono state riscontrate problematiche.



In particolare, “Mission Critical” indica che l’alert rilevato deriva da target al quale prestare maggiore riguardo poiché può comportare l’interruzione del business; “Production” riferisce ad incidents di livello intermedio che non provocano interruzioni del business ma piuttosto su strumenti di lavoro interni all’azienda; infine “Development Test” allude ad incidents su target di minore criticità. Nel momento in cui viene rilevato un evento da parte del sistema di anomaly detection, quello stesso evento sarà indirizzato in una delle tre aree in base alla sua classificazione. Una volta stabilita l’etichetta dell’evento, quest’ultimo verrà raggruppato nelle sotto aree in base al suo livello di criticità (“FATAL”, “CRITICAL” oppure “WARNING”) e alla categoria di appartenenza che sta ad indicare se l’allarme deriva da problemi connessi alla disponibilità di servizi sui sistemi clienti (“Availability”), se è connesso all’occupazione di componenti quali CPU, memoria etc. (“Capacity”), oppure se riguarda il backup dei database dei clienti (“Backup”). La sezione “Altro” verrà meglio indagata nel paragrafo ad essa dedicata ed anche nel terzo capitolo dell’elaborato, inerente agli sviluppi futuri dell’applicazione.

## 2.2.1 Mission Critical

La sezione “Mission Critical” è stata generata inserendo una region all’interno del body della pagina Panoramica attraverso il Page Designer. Dall’analisi dei requisiti, è risultato che la tipologia di region, offerta dall’IDE di APEX, che meglio sposava le desiderate modalità di visualizzazione e gestione dei dati, fosse un Classic Report mostrato con template Media List. Questo implica che il report non mostrerà i dati nel formato tabellare standard, ma piuttosto seguirà il design pattern del template Media List; quest’ultimo consente, infatti, di mostrare le molteplici informazioni di un dato attraverso titoli, descrizioni, icone e badges. Per una maggiore comprensione, l’immagine sottostante mostra un esempio di classic report con template Media List.

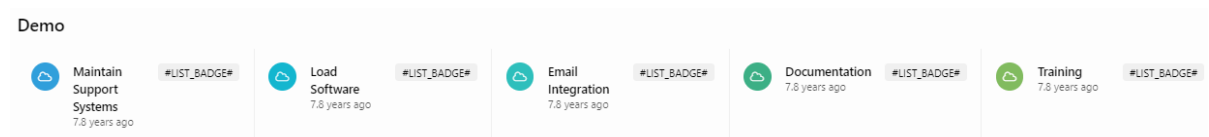
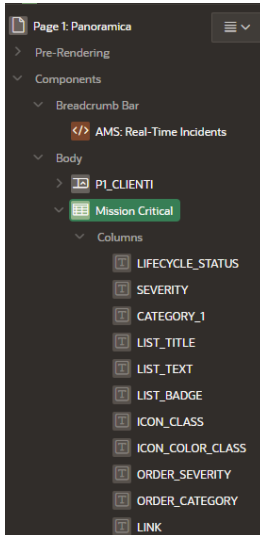


FIGURA 32: MEDIA LIST TEMPLATE<sup>6</sup>

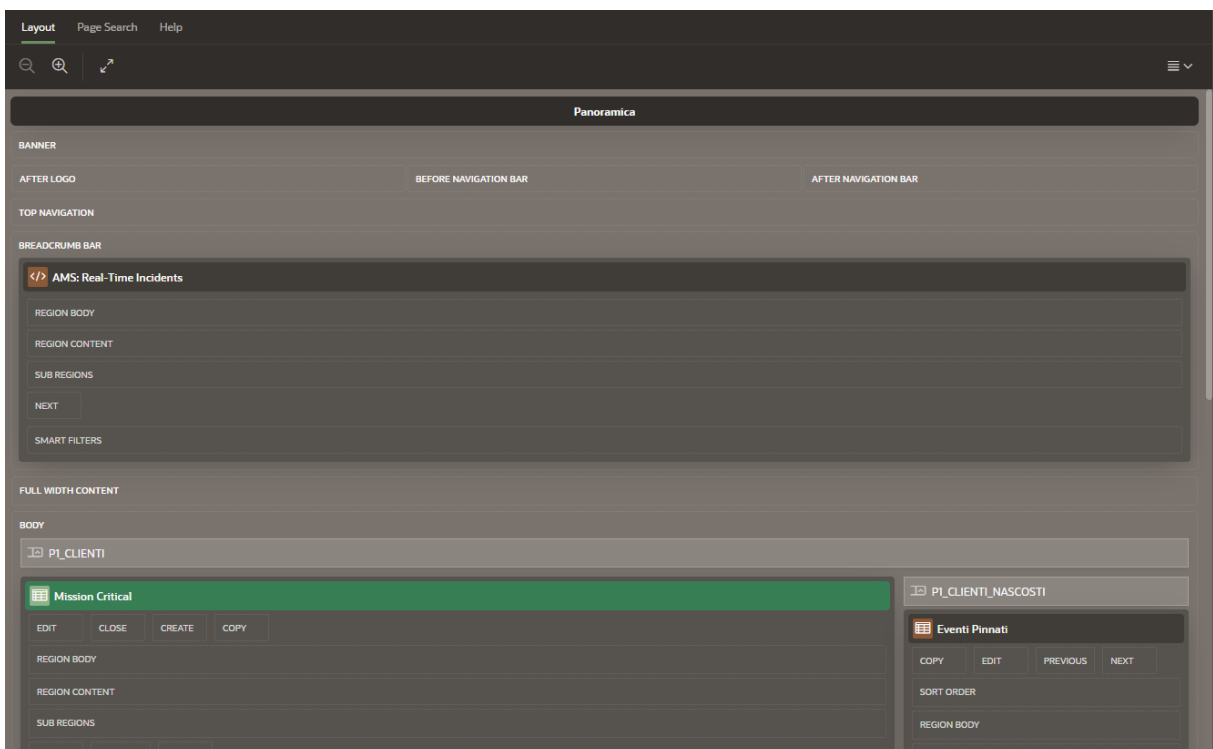
Per comprendere le logiche di quanto visualizzato nell’esecuzione della pagina Panoramica, occorre rivolgere l’attenzione verso il Page Designer; il Tree Pane, mostra che Mission Critical

<sup>6</sup> [https://apex.oracle.com/pls/apex/r/apex\\_pm/ut/getting-started](https://apex.oracle.com/pls/apex/r/apex_pm/ut/getting-started)

è dipendente da alcune colonne, di cui a breve si darà un significato, mentre il pannello centrale offre un'anteprima della pagina. Le figure sottostanti rappresentano rispettivamente il pannello sinistro e quello centrale.



**FIGURA 33: MISSION CRITICAL TREE PANE**



**FIGURA 34: LAYOUT DELLA SEZIONE MISSION CRITICAL**

Invece, tutte le proprietà relative alla Region sono localizzate nel Property Editor, la sezione destra del Page Designer, mostrato nella figura 35 riportata di seguito.

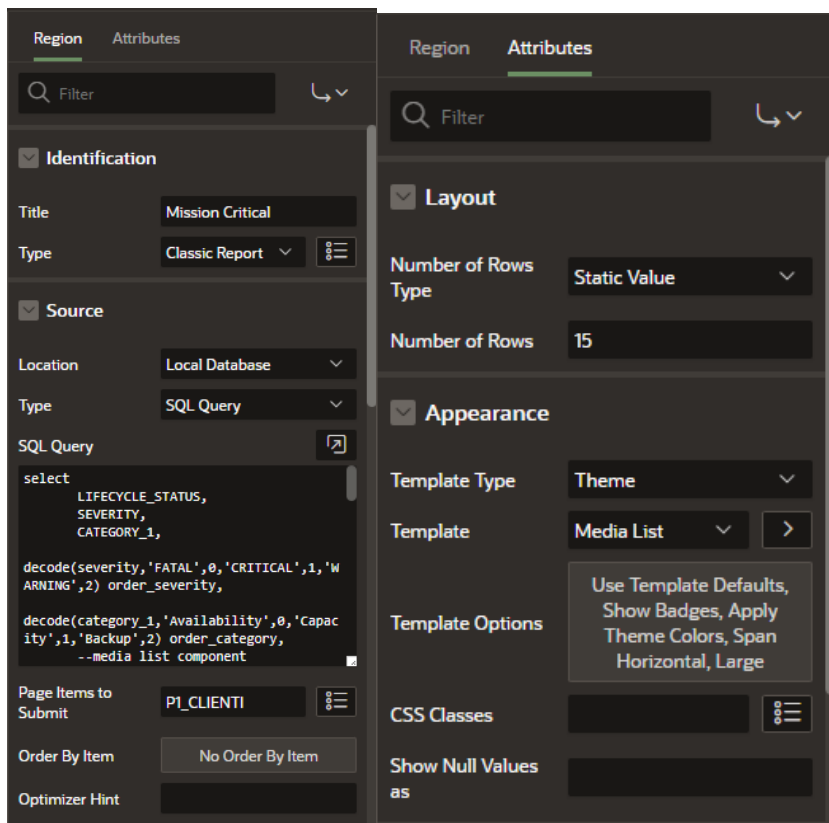


FIGURA 35: MISSION CRITICAL PROPERTY EDITOR

La proprietà “Identification” richiede di inserire il titolo e la modalità di rendering della region, in questo caso è stato preferito il Classic Report; “Source” indica la sorgente dati, che è un database locale e la tipologia di ritorno dei dati che avviene tramite query SQL. “Page Item to Submit” è un attributo mediante il quale si stabilisce che tutti i dati della region siano filtrati per il page item indicato, in questo caso “P1\_CLIENTI”. Nella scheda “Attributes”, il template Media List presenta una serie di impostazioni che consentono di mostrare badges, di espandere la region orizzontalmente etc.

È opportuno focalizzare l’attenzione sulla query che regola la Media List “Mission Control”;

```

Code Editor - SQL Query
1 select
2     LIFECYCLE_STATUS,
3     SEVERITY,
4     CATEGORY_1,
5     decode(severity,'FATAL',0,'CRITICAL',1,'WARNING',2) order_severity,
6     decode(category_1,'Availability',0,'Capacity',1,'Backup',2) order_category,
7
8     --Media List
9     CATEGORY_1 LIST_TITLE,
10    SEVERITY LIST_TEXT,
11    COUNT(*)-sum(nvl((select 1 from all_notification_ignore where message like '%'||message_pattern||'%'),0)) LIST_BADGE,
12    apex_page.get_url(p_page => 2, p_items => 'P2_LIFECYCLE_STATUS,P2_SEVERITY,P2_CATEGORY_1,P2_CLIENTE' , p_values =>LIFECYCLE_STATUS||','||SEVERITY||','||CATEGORY_1||','||P2_CLIENTE) LINK ,
13    (case
14        when severity = 'WARNING' then 'fa fa-warning fa-lg'
15        when severity = 'CRITICAL' then 'fa fa-exclamation-circle fa-lg'
16        when severity = 'FATAL' then 'fa fa-fire fa-2x fa-lg'
17        end) ICON_CLASS,
18    (case when severity = 'WARNING' then 'u-color-7'
19
20        when severity = 'CRITICAL' then 'u-color-9'
21
22        when severity = 'FATAL' then 'u-color-15'
23
24        end) ICON_COLOR_CLASS
25
26
27 from INCIDENT
28 where lifecycle_status = 'MissionCritical'
29 and hidden = 0 --$filter_line_of_business
30 and category_1 in ('Availability','Capacity','Backup')
31 and severity in ('FATAL','CRITICAL','WARNING')
32 and NOT EXISTS (SELECT 1 FROM clienti WHERE nome = line_of_business AND hidden_cliente=1)
33 and (line_of_business = :P1_CLIENTE or :P1_CLIENTE is null)
34 group by LIFECYCLE_STATUS,SEVERITY,CATEGORY_1
35 order by order_category,order_severity;

```

FIGURA 36: QUERY SQL CHE REGOLA MISSION CONTROL

Le istruzioni dell'interrogazione SQL mostrata in figura 36, richiedono di restituire dalla tabella Incident, i campi *lifecycle\_status*, *severity* e *category\_1* di ogni evento con il vincolo che *lifecycle\_status* abbia valore uguale a "MissionCritical"; si vuole che *severity*, ossia il livello di criticità dell>alert, sia unicamente "FATAL", "CRITICAL" oppure "WARNING" e *category\_1*, ossia la categoria, sia di tipo "Availability", "Capacity" oppure "Backup". È necessario raggruppare i dati per *lifecycle\_status*, *severity*, *category\_1*, mediante lo statement `group by`, ed ordinarli per categoria e severity, mediante statement `order by`. L'order by consente di ordinare gli incident dando priorità alla categoria e per ciascuna categoria, mostrare dapprima gli eventi con criticità "FATAL", poi "CRITICAL" ed infine "WARNING". Quanto detto è riscontrabile nell'immagine, dapprima riportata, raffigurante la page Panoramica. Nell'istruzione `select`, viene inoltre gestita la Media List, più precisamente, viene indicato cosa attribuire ai suoi componenti: `LIST_TITLE` è il titolo, rappresentato in questo caso dalla categoria dell'incident; `LIST_TEXT` è la descrizione, attribuita alla severity; `LIST_BADGE` è un contatore che mostrerà il numero di tutti gli incident, di una precisa categoria e criticità, eccetto quelli che hanno un pattern presente nella tabella dati *all\_notification\_ignore*, che indica appunto messaggi da ignorare. `LINK` consente di indirizzare l'utente ad un'altra pagina, tramite un semplice click, mediante la funzione `apex_page.get_url()` che richiede di specificare la pagina a cui si vuole giungere ed i page items lì presenti, ai quali si vogliono associare i valori specificati in `p.values`. In questo caso, si richiede di portare i valori di *lifecycle\_status*, *severity*,

*category\_1* e *P1\_CLIENTI* negli item di pagina 2 denominati *P2\_LIFECYCLE\_STATUS*, *P2\_SEVERITY*, *P2\_CATEGORY\_1* e *P2\_CLIENTE*; *ICON\_CLASS* consente di indicare le icone da attribuire in base ai valori della severity; *ICON\_COLOR\_CLASS* gestisce la selezione del colore da attribuire alle icone. Nello statement where della query è inoltre presente la condizione *line\_of\_business =:P1\_CLIENTI or :P1\_CLIENTI is null*, che indica che il valore di *line\_of\_business*, ovvero il nome del cliente, sia uguale al valore del page item *P1\_CLIENTI* oppure nullo. *P1\_CLIENTI* è un page item di tipo *PopupLOV*, visibile nel *Tree Pane*, nel pannello centrale e nella pagina *Panoramica* renderizzata, sopra “*Mission Control*”. Una *PopupLOV*, ha l’aspetto di una barra di ricerca che, se oggetto di selezione, mostra una finestra con una lista di valori. In questo caso, mostrerà la lista dei clienti aziendali e, mediante la where condition appena citata, si stabilisce che se un cliente viene selezionato, verranno mostrati nella sezione *Mission Control*, ma anche in *Production* e in *Development Test*, solo gli allarmi provenienti da quel preciso cliente; se nessun cliente verrà selezionato dalla *PopupLOV*, nessun filtraggio per cliente verrà effettuato.

L’immagine 37 mostra il ramo del *Tree Pane* riferito al page item “*P1\_CLIENTI*”: in particolare si vuole evidenziare che è stata creata una *Dynamic Action* “*onChangeClienti*”, che si attiva nel momento in cui si verifica un evento di tipo “*Change*”, in questo caso un cambiamento al valore dell’item “*P1\_CLIENTI*”. Ciò che consegue dall’attivazione di “*onChangeClienti*”, è l’esecuzione di quattro azioni di tipo “*Refresh*”, una per ciascuna region interessata, ossia “*Mission Critical*”, “*Production*”, “*Development Test*” e “*Altro*”. In altre parole, nel momento in cui un cliente viene selezionato dalla *PopupLOV* (evento “*Change*”), le quattro *Media Lists* saranno soggette alla *Dynamic Action* “*onChangeClienti*”; per cui se “*onChangeClienti*” viene innescata (“*True*”), le quattro sezioni interessate saranno oggetto di *refresh*, in modo da mostrare solo le informazioni degli incident relative al cliente selezionato nella *PopupLOV*.

Quanto appena descritto, può essere riscontrato nelle immagini successive che mostrano rispettivamente, la scheda di *Rendering* con focus sul page item “*P1\_CLIENTI*” e le relative *Dynamic Actions*, la query che regola “*P1\_CLIENTI*” ed infine la *PopupLOV* nella page *Panoramica*.

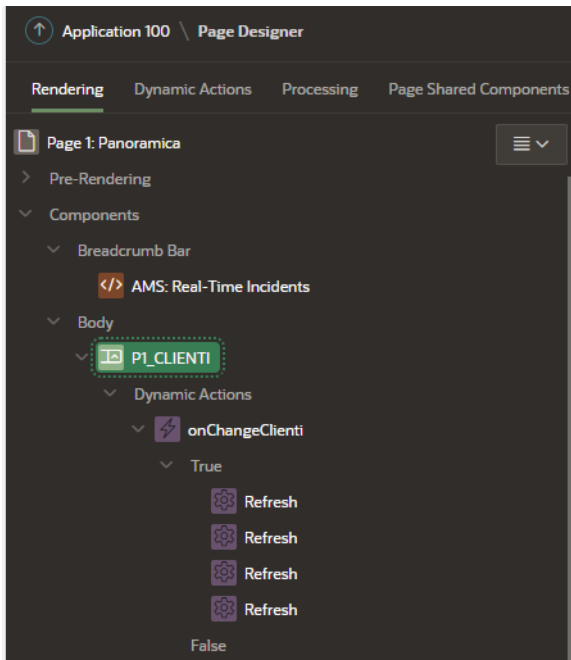


FIGURA 37: PAGE ITEM "P1\_CLIENTI"

```

1 select distinct line_of_business d
2 from incident
3 where hidden = 0 --$filter_line_of_business
4     and category_1 in ('Availability','Capacity','Backup')
5     and severity in ('FATAL','CRITICAL','WARNING')
6     and NOT EXISTS (SELECT 1 FROM clienti WHERE nome = line_of_business AND hidden_cliente=1);
7

```

FIGURA 38: QUERY SQL CHE REGOLA "P1\_CLIENTI"

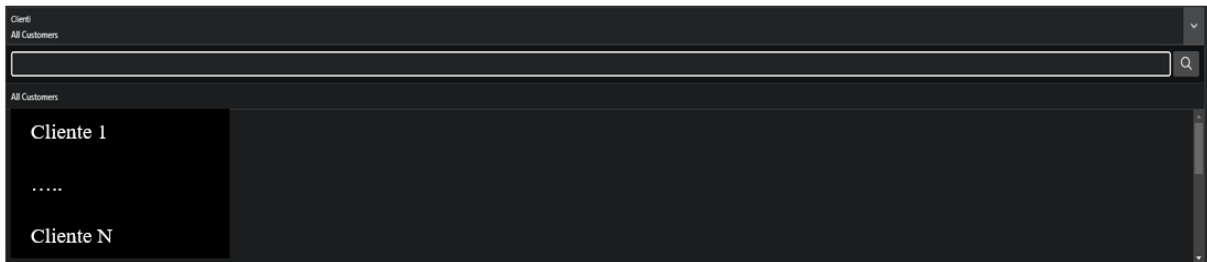


FIGURA 39: POPUPLOV CLIENTI

## 2.2.2 Production

Questa sezione, renderizzata anch'essa mediante Classic Report con template Media List, differisce da "Mission Critical" semplicemente per una condizione nello statement where della query che consente di mostrare gli incidents.

```

Code Editor - SQL Query
1 select
2   LIFECYCLE_STATUS,
3   SEVERITY,
4   CATEGORY_1,
5   decode(severity,'FATAL',0,'CRITICAL',1,'WARNING',2) order_severity,
6   decode(category_1,'Availability',0,'Capacity',1,'Backup',2) order_category,
7   --media list
8   CATEGORY_1 LIST_TITLE,
9   SEVERITY LIST_TEXT,
10  COUNT(*)-sum(nvl((select 1 from all_notification_ignore where message like '%'||message_pattern||'%'),0)) LIST_BADGE,
11  apex_page.get_url(p_page => 2, p_items => 'P2_LIFECYCLE_STATUS,P2_SEVERITY,P2_CATEGORY_1,P2_CLIENTE' , p_values =>LIFECYCLE_STATUS||','||SEVERITY||','||CATEGORY_1||','||:P1_CLIENTI LINK ,
12  (case
13    when severity = 'WARNING' then 'fa fa-warning fa-lg'
14
15
16    when severity = 'CRITICAL' then 'fa fa-exclamation-circle fa-lg'
17    when severity = 'FATAL' then 'fa fa-fire fa-2x fa-lg'
18
19    end) ICON_CLASS,
20  (case when severity = 'WARNING' then 'u-color-7'
21    when severity = 'CRITICAL' then 'u-color-9'
22    when severity = 'FATAL' then 'u-color-15'
23    end) ICON_COLOR_CLASS
24
25
26 from INCIDENT
27 where lifecycle_status = 'Production'
28 and hidden = 0 --filter line_of_business
29 and category_1 in ('Availability','Capacity','Backup')
30 and severity in ('FATAL','CRITICAL','WARNING')
31 and NOT EXISTS (SELECT 1 FROM clienti WHERE nome = line_of_business AND hidden_cliente=1)
32 and (line_of_business = :P1_CLIENTI or :P1_CLIENTI is null)
33 group by LIFECYCLE_STATUS,SEVERITY,CATEGORY_1
34 order by order_category,order_severity
35
Cancel OK

```

FIGURA 40: QUERY CHE REGOLA PRODUCTION

Come si può notare dall'immagine, l'unica differenza è nella where condition, in cui si impone di mostrare gli incident per cui *lifecycle\_status* abbia valore pari alla stringa 'Production'.

### 2.2.3 Development Test

Questa region differisce dalle due precedenti, semplicemente per il fatto che "Development" e "Test" sono due differenti etichette, valori di *lifecycle\_status*. Per unirli e mostrare gli incidents con entrambe le etichette nella stessa sezione, è stata leggermente modificata la query SQL che regola anche "Production" e "Mission Control".

```

Code Editor - SQL Query
1 select
2 decode(lifecycle_status,'Test','Development Test','Development','Development Test',lifecycle_status) lifecycle_status,
3 SEVERITY,
4 CATEGORY_1,
5 decode(severity,'FATAL',0,'CRITICAL',1,'WARNING',2) order_severity,
6 decode(category_1,'Availability',0,'Capacity',1,'Backup',2) order_category,
7 -- media list component
8 CATEGORY_1 LIST_TITLE,
9 SEVERITY LIST_TEXT,
10 COUNT(*)-sum(nvl((select 1 from all_notification_ignore where message like '%||message_pattern||%' ),0)) LIST_BADGE,
11 apex_page.get_url(p_page => 2, p_items => 'P2_LIFECYCLE_STATUS,P2_SEVERITY,P2_CATEGORY_1,P2_CLIENTE' , p_values =>
12 decode(lifecycle_status,'Test','Development Test','Development','Development Test',lifecycle_status)||','||SEVERITY||','||CATEGORY_1||','||:P1_CLIENTI LINK ,
13 (case
14 when severity = 'WARNING' then 'fa fa-warning fa-lg'
15
16
17 when severity = 'CRITICAL' then 'fa fa-exclamation-circle fa-lg'
18 when severity = 'FATAL' then 'fa fa-fire fa-2x fa-lg'
19
20 end) ICON_CLASS,
21 ((case when severity = 'WARNING' then 'u-color-7' |
22 when severity = 'CRITICAL' then 'u-color-9'
23 when severity = 'FATAL' then 'u-color-15'
24 end)) ICON_COLOR_CLASS
25 from INCIDENT
26
27 where lifecycle_status in ('Test','Development')
28 and hidden = 0 --$filter_line_of_business
29 and category_1 in ('Availability','Capacity','Backup')
30 and severity in ('FATAL','CRITICAL','WARNING')
31 and NOT EXISTS (SELECT 1 FROM clienti WHERE nome = line_of_business AND hidden_cliente=1)
32 and (line_of_business = :P1_CLIENTI or :P1_CLIENTI is null)
33 group by decode(lifecycle_status,'Test','Development Test','Development','Development Test',lifecycle_status),SEVERITY,CATEGORY_1
34 order by order_category,order_severity;

```

FIGURA 41: QUERY CHE REGOLA DEVELOPMENT TEST

La nuova condizione where per *lifecycle\_status* impone che quest'ultimo abbia valori che siano Test o Development; inoltre, per unire le due etichette in un'unica, viene utilizzata l'istruzione decode la cui semantica è la seguente: se *lifecycle\_status* ha valore 'Test', viene sostituito con 'Development Test', se ha valore 'Development' viene sostituito comunque con 'Development Test', se non ha nessuno dei due valori, ritorna il valore corrente. È necessario pertanto riportare la decode sia nella group by che all'interno della funzione *apex\_page.get\_url()*, per inviare alla pagina 2 un valore di *lifecycle\_status* pari a 'Development Test' da affidare al page item *P2\_LIFECYCLE\_STATUS*.

## 2.2.4 Altro

Quest'area della pagina "Panoramica" è gestita anch'essa con il componente APEX di tipo Classic Report con template Media List i cui elementi hanno semantiche distinte ed indirizzano a differenti pagine dell'applicazione. Per poter approfondire ciascuno di essi, è opportuno analizzare la query che gestisce l'intera sezione visibile nelle figure x e x.



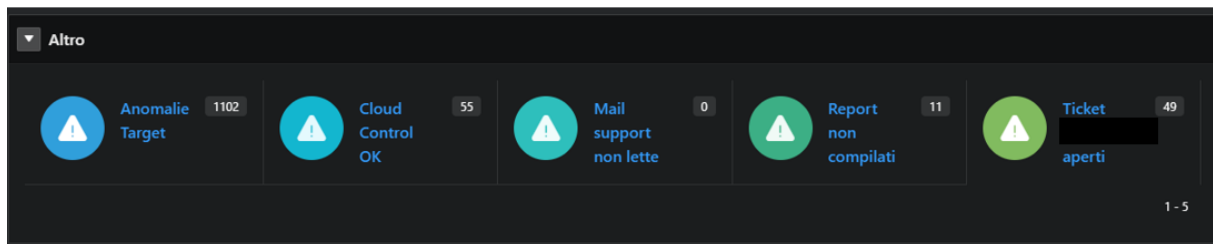


FIGURA 42: ALTRO

```

1 SELECT
2 'Anomalia Target' LIST_TITLE,
3 '' LIST_TEXT,
4 COUNT(*) LIST_BADGE,
5
6 case
7 when count(*) > 0 then 'fa fa-warning fa'
8 when count(*) = 0 then 'fa fa-check fa'
9 end ICON_CLASS,
10
11 case
12 when count(*)>0 then 'u-warning'
13 when count(*)=0 then 'u-color-35'
14 end ICON_COLOR_CLASS,
15 apex_page.get_url(p_page => 3 ) LINK
16 FROM(
17 SELECT * FROM mm_ams.target_saved WHERE lifecycle_status IS NULL
18 OR lifecycle_status='
19 OR (cloud_control_id=1 AND line_of_business IS NULL)
20 OR (cloud_control_id=1 AND line_of_business ='')
21 )
22 WHERE 1=1
23 UNION
24 SELECT 'Cloud control OK' LIST_TITLE,
25 '' LIST_TEXT,
26 COUNT(*) LIST_BADGE,
27
28 case
29 when count(*) > 0 then 'fa fa-warning '
30 when count(*) = 0 then 'fa fa-check '
31 end ICON_CLASS,
32
33 case
34 when count(*)>0 then 'u-warning'
35 when count(*)=0 then 'u-color-35'
36 end ICON_COLOR_CLASS,
37 apex_page.get_url(p_page => 4 ) LINK
38 FROM cloud_control
39 WHERE dismissed=0
40 UNION
41 SELECT 'Mail support non lette' LIST_TITLE,
42 '' LIST_TEXT,
43 sum(valore) LIST_BADGE,
44
45 case
46 when count(*) > 0 then 'fa fa-warning '
47 when count(*) = 0 then 'fa fa-check '
48 end ICON_CLASS,
49
50 case
51 when count(*)>0 then 'u-warning'
52 when count(*)=0 then 'u-color-35'
53 end ICON_COLOR_CLASS,
54 apex_page.get_url(p_page => 3 ) LINK
55 FROM mm_ams.contattori
56 WHERE chiave='email_support_non_lette'
57
58 UNION
59 SELECT 'Ticket ████████ aperti' LIST_TITLE,
60 '' LIST_TEXT,
61 sum(to_number(value)) LIST_BADGE,
62
63 case
64 when count(*) > 0 then 'fa fa-warning '
65 when count(*) = 0 then 'fa fa-check '
66 end ICON_CLASS,
67
68 case
69 when count(*)>0 then 'u-warning'
70 when count(*)=0 then 'u-color-35'
71 end ICON_COLOR_CLASS,
72 FROM mm_ams.incident ot
73 where event_name like 'ME$NuoveRichieste:Numero_Ticket'
74 UNION
75 SELECT 'Report non compilati' LIST_TITLE,
76 '' LIST_TEXT,
77 COUNT(*) LIST_BADGE,
78
79 case
80 when count(*) > 0 then 'fa fa-warning '
81 when count(*) = 0 then 'fa fa-check '
82 end ICON_CLASS,
83
84 case
85 when count(*)>0 then 'u-warning'
86 when count(*)=0 then 'u-color-35'
87 end ICON_COLOR_CLASS,
88 apex_page.get_url(p_page => 6 ) LINK
89 FROM mm_ams.clienti_no_report
90 WHERE 1=1 ;

```

FIGURA 43: QUERY CHE REGOLA LA SEZIONE ALTRO



da un region di tipo Interactive Report, la cui sorgente dati è rappresentata dalla tabella “TARGET\_SAVED”; i dati della tabella visualizzati saranno quelli sottoposti alla condizione where prima descritta, ossia che hanno campo *lifecycle\_status* con valore NULL o vuoto, oppure con campo *cloud\_control\_id* =1 ed il valore NULL oppure vuoto di *line\_of\_business*. L’immagine 45 mostra la scheda region del report nel Property Editor.

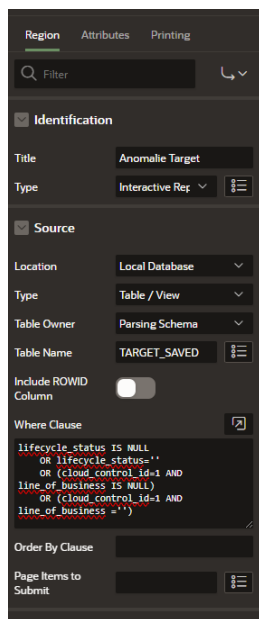


FIGURA 45: ANOMALIE TARGET NEL PROPERTY EDITOR

- **Cloud Control OK:** elemento della Media List finalizzato a mostrare se il Cloud Control, ossia il componente denominato Enterprise Manager che rileva gli incidents dai sistemi dei clienti (mostrati in AMS:Real-Time Incidents), sta operando correttamente. I dati restituiti dalla query che riguarda “Cloud Control OK” provengono dalla tabella *cloud\_control*. Nella region Media List vengono mostrati “Cloud Control OK” come LIST\_TITLE, una stringa vuota per l’elemento LIST\_TEXT e il numero dei dati nella table *cloud\_control* come LIST\_BADGE; due case statements gestiscono l’ICON\_CLASS e l’ICON\_COLOR\_CLASS, mentre per l’elemento LINK viene utilizzata la funzione *apex\_page.get\_url(p.page=>4)* per collegare l’utente con la pagina 4 alla pressione della sezione “Cloud Control OK”. La pagina 4 coincide con la pagina “Cloud Control” presente nel menu di navigazione della pagina Panoramica e di cui si indagherà meglio in seguito.
- **Mail Support Non Lette:** è legata ad una funzionalità dell’applicazione AMS:Real-Time Incidents che non è stata ancora implementata e verrà esaminata nel terzo capitolo.

- **Ticket ClienteX aperti:** questo elemento della Media List non indirizza ad alcuna pagina, come si può vedere nella query; esso ha la funzione di mostrare, tramite l'elemento LIST\_BADGE, il numero dei ticket che sono stati creati per il ClienteX, calcolati prelevandoli dalla tabella incident dove il campo event\_name contiene la stringa 'ME\$NuoveRichieste:Numero\_Ticket'. Più nello specifico, un ticket ha la funzione di mostrare le informazioni relative ad un allarme di un cliente, da chi è stato gestito, eventuali note da e verso il cliente etc. Per il ClienteX non è prevista l'implementazione del servizio di ticketing, distintamente da altri clienti. Il servizio di ticketing non è stato ancora implementato, ma verrà analizzato nell'ultimo capitolo.
- **Report non compilati:** ha la funzione di indirizzare ad una pagina denominata “Clienti No Report” mediante l'elemento LINK della Media List tramite istruzione `apex_page.get_url(p_page => 6)`, che mostra un report con i nomi dei clienti per cui non è stato ancora stilato un report mensile con eventuali proposte destinate al cliente. La query prevede di prelevare dalla tabella clienti\_no\_report il numero di clienti (elemento LIST\_BADGE) che non hanno realizzato tale report mensile; all'elemento LIST\_TITLE è associata la stringa “Report non compilati”, mentre due case statements gestiscono l'icona e il colore dell'icona. Cliccando su “Report non compilati”, l'utente viene reindirizzato ad una pagina denominata “Clienti No Report” che contiene una region di tipo classic report con la lista dei clienti presenti nella tabella CLIENTI\_NO\_REPORT.

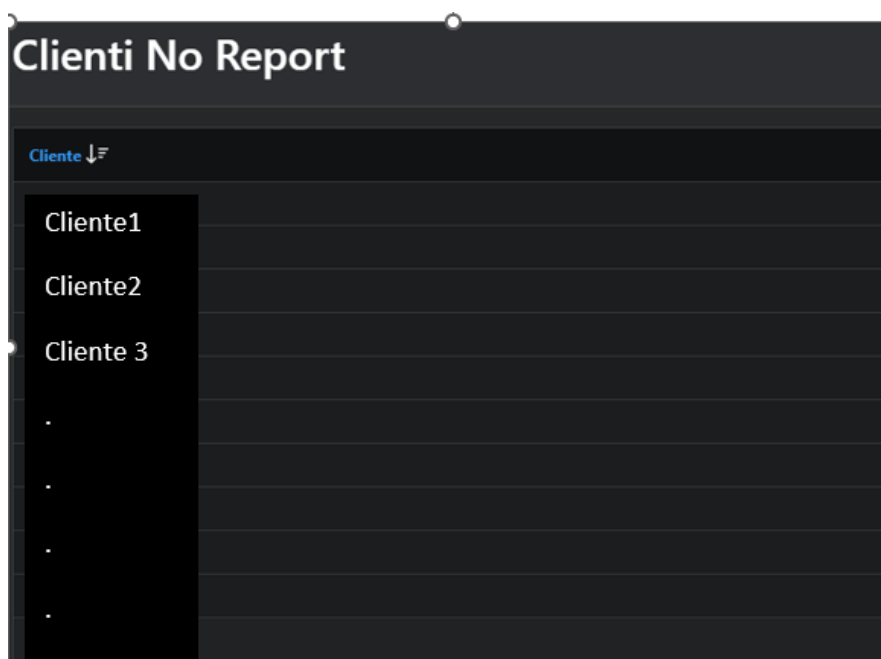


FIGURA 46: REPORT CLIENTI PER CUI NON È STATO STILATO IL REPORT MENSILE

## 2.2.5 Eventi Pinnati

Come precedentemente anticipato e mostrato nell'immagine raffigurante l'home page "Panoramica", la sezione destra è abitata da due regions, di cui una è "Eventi Pinnati". Quest'area mostra tutti gli incidents messi in evidenza dagli utenti; le tecniche che consentono il pin, l'azione di mettere in rilievo un evento, ed anche lo spin, ossia la rimozione dell'evento dal report, verranno ben approfondite nel corso del paragrafo. Il pannello sinistro del page designer mostra la region "Eventi Pinnati" con le relative colonne, risultato della query elaborata nel Property Editor, ed inoltre una Dynamic Action "DialogClosedAfterSpin", che verrà presto indagata.

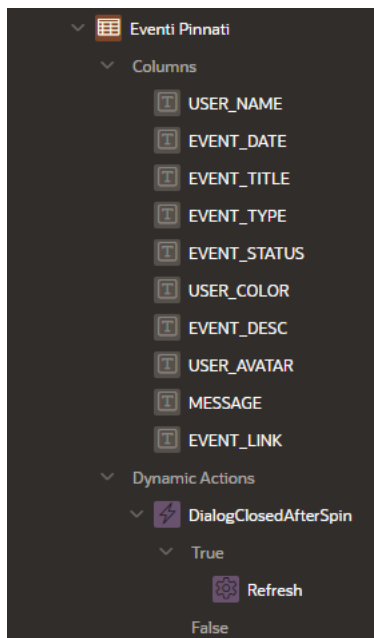


FIGURA 47: TREE PANE DELLA REGION EVENTI PINNATI

Dall'analisi dei requisiti è emerso che, il componente APEX che meglio approssimasse l'idea di visualizzazione di un incident e che riassume le principali caratteristiche fosse un Classic Report con template Activity Timeline. Generalmente una Timeline viene utilizzata per mostrare una serie di eventi o aggiornamenti recenti in real-time, in questo caso gli eventi rilevati e messi in evidenza dagli end users.

Nel Property Editor del Page Designer, la region "Eventi Pinnati" è regolata da una query SQL ed il template selezionato è Timeline con modalità di visualizzazione compatta.

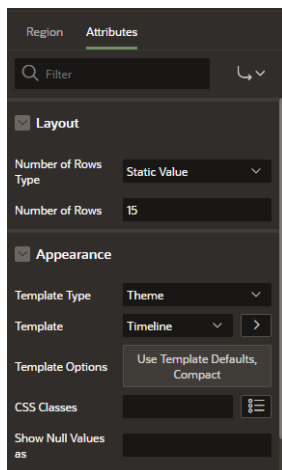


FIGURA 48: PROPERTY EDITOR DI EVENTI PINNATI

Per comprendere al meglio la query SQL che consente di visualizzare gli eventi pinnati, dapprima occorre comprendere com'è strutturata una Timeline ed i suoi elementi:

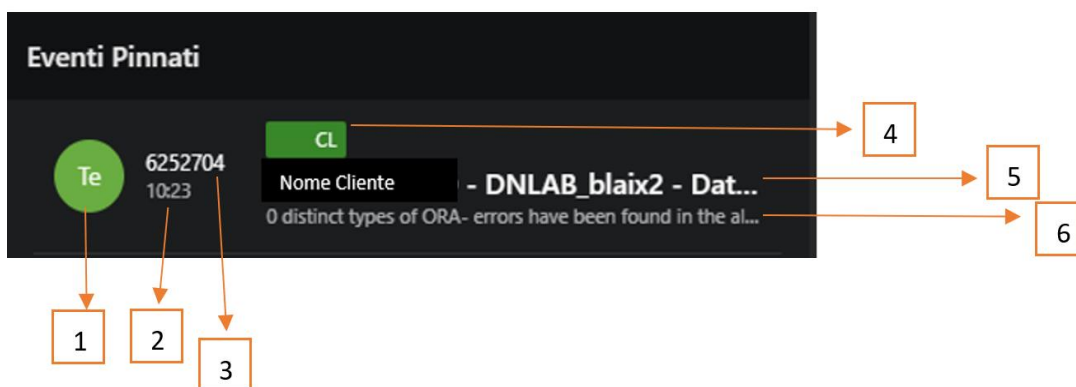


FIGURA 49: ACTIVITY TIMELINE DI UN EVENTO

- Il componente “1” coincide con l’elemento USER\_AVATAR di una Timeline, finalizzato a mostrare le iniziali di un dato, il cui colore viene regolato mediante USER\_COLOR.
- il componente “2” è associato all’elemento EVENT\_DATE.
- il componente “3” è riservato allo USER\_NAME.
- il componente “4” è l’EVENT\_TYPE, il cui colore viene gestito mediante EVENT\_STATUS.
- il componente “5” è l’EVENT\_TITLE, che tipicamente offre informazioni circa il nome dell’evento.
- il componente “6” è l’EVENT\_DESC che contiene generalmente una descrizione dell’evento.

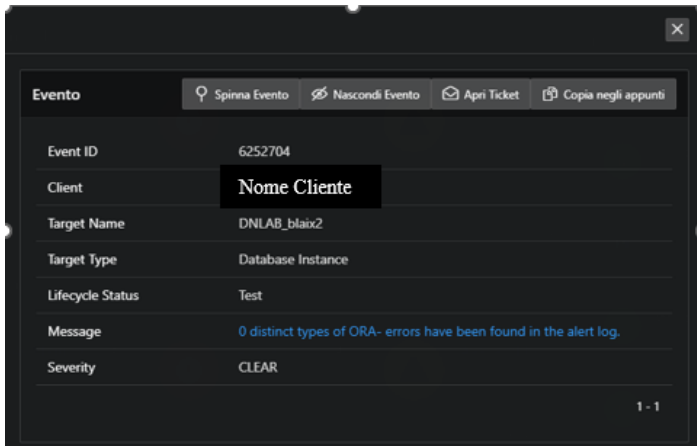
- il template Activity Timeline consente di indirizzare l'utente ad un'altra pagina tramite l'elemento EVENT\_LINK.

```
Code Editor - SQL Query
1 select
2     i.message ,
3     --Timeline Activity
4     i.mm_event_id USER_NAME,
5     to_char(ip.pinned_time,'HH24:MI') EVENT_DATE,
6     UPPER(substr(i.severity,1,2)) EVENT_TYPE,
7     (i.line_of_business||' - '||i.target_name||' - '|| i.target_type) EVENT_TITLE,
8     case
9         when i.severity='WARNING' then 'u-warning'
10        when i.severity='FATAL' then 'u-color-15'
11        when i.severity='CRITICAL' then 'u-danger'
12        when i.severity='CLEAR' then 'u-success'
13    end EVENT_STATUS,
14    substr(i.lifecycle_status,1,2) USER_AVATAR,
15    case
16        when i.lifecycle_status='MissionCritical' then 'u-color-15'
17        when i.lifecycle_status='Production' then 'u-color-24'
18        when i.lifecycle_status='Test' then 'u-color-35'
19        when i.lifecycle_status='Development' then 'u-color-35'
20    end USER_COLOR ,
21    apex_page.get_url(p_page => 7, p_items => 'P7_MM_EVENT_ID', p_values => i.mm_event_id) EVENT_LINK,
22    message EVENT_DESC
23 from incident i , incident_pinnati ip
24 where i.mm_event_id = ip.mm_event_id
25        and i.pinned=1
26        and i.hidden=0;
```

FIGURA 50: QUERY SQL CHE REGOLA EVENTI PINNATI

L'interrogazione SQL appena mostrata, richiede di prelevare le informazioni da due tabelle, rispettivamente, "incident" ed "incident\_pinnati"; la prima contiene tutte le informazioni relative ad un evento rilevato, mentre la seconda include l'identificativo di un evento e l'informazione "pinned=0" se l'event non è stato posto in evidenza, in caso contrario "pinned=1". Eseguendo una join delle due tabelle tramite chiave primaria *mm\_event\_id* verranno selezionati solo gli eventi con pinned=1 e quelli non nascosti ("hidden=0"). Tutti gli elementi della Timeline prima introdotti, assumono nella query dei valori ben precisi: l'elemento USER\_AVATAR conterrà le iniziali di *lifecycle\_status*, ovvero l'etichetta che può assumere valori come "Mission Critical", "Production", etc. ; USER\_NAME corrisponderà all'ID dell'incident; EVENT\_DATE è associato al "pinned\_time", l'orario in cui si è verificato il "pin" dell'evento da parte dell'utente; EVENT\_TYPE corrisponde alla criticità (severity) dell>alert; EVENT\_TITLE è il risultato della concatenazione delle stringhe *line\_of\_business* (nome del cliente), *target\_name* (la tipologia del problema) e *target\_type* (un'ulteriore informazione inerente alla tipologia del problema); EVENT\_DESC è il messaggio descrittivo dell'evento; EVENT\_LINK regolato dalla funzione *apex\_page.get\_url()* consente di indirizzare l'user alla pagina 7 associando il valore dell'id dell'incident a "P7\_MM\_EVENT\_ID" (page item di pagina 7). Infatti, cliccando un evento della Timeline,

l'utente verrà introdotto in una nuova pagina dell'applicazione, in modal dialog, che consente di visualizzare le informazioni relative all'evento più dettagliatamente ed inoltre generare una serie di azioni quali, lo “spin” dell'evento, nascondere l'evento, aprire un ticket oppure copiare i dettagli dell'incident, il tutto mediante l'utilizzo di appositi pulsanti. L'immagine che segue raffigura quanto detto.



**FIGURA 51: DETTAGLI DI UN EVENTO IN EVIDENZA**

La region “Evento” è di tipo Classic Report con template Value Attribute Pairs-Column, che consente di restituire i dati risultanti dalla query in forma “Nome Colonna della tabella”-“Valore”. In questo caso l'interrogazione che ritorna i dati da mostrare è la seguente:

```

1 select MM_EVENT_ID,
2        INCIDENT_ID,
3        SEQUENCE_ID,
4        CLOUD_CONTROL_ID,
5        INCIDENT_STATUS,
6        LINE_OF_BUSINESS,
7        CATEGORY_1,
8        TARGET_NAME,
9        TARGET_TYPE,
10       HOSTNAME,
11       LIFECYCLE_STATUS,
12       EVENT_TYPE,
13       EVENT_NAME,
14       MESSAGE,
15       SEVERITY,
16       VALUE,
17       CREATED,
18       UPDATED,
19       LAST_UPDATE_CC,
20       HIDDEN,
21       PINNED,
22       TICKET_OPENED,
23       SEVERITYPOINTS
24 from INCIDENT
25 where mm_event_id = :P7_MM_EVENT_ID

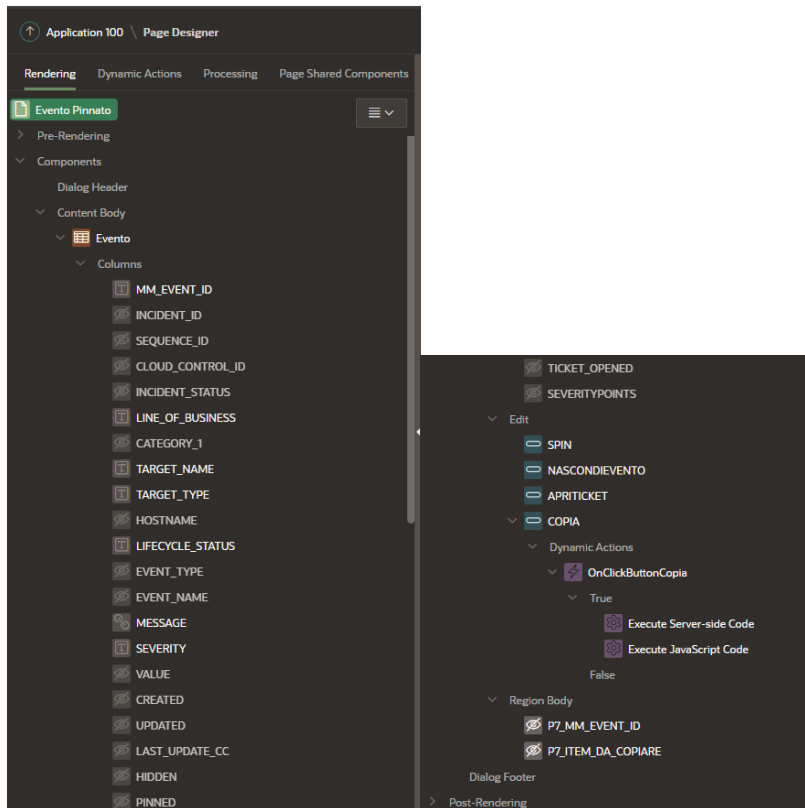
```

**FIGURA 52: QUERY CHE REGOLA IL REPORT DI UN EVENTO**

Vengono selezionati tutti i campi della table “Incident” dove l'identificativo dell'evento “mm\_event\_id” assume il valore del page item P7\_MM\_EVENT\_ID, che ricordiamo assumere



valore tramite la funzione `apex_page.get_url(p_page => 7, p_items => 'P7_MM_EVENT_ID', p_values => i.mm_event_id) EVENT_LINK`, presente nella query che gestisce la Timeline “Eventi Pinnati”. Non tutti i campi presenti nella query verranno mostrati nel report; ciò è verificabile ispezionando la scheda di Rendering del Tree Pane.



**FIGURA 53: TREE PANE DELLA PAGINA EVENTO PINNATO**

Come si può notare, non tutte le colonne sono impostate come visibili, alcune saranno di tipo Plain Text, altre di tipo Hidden ed una in particolare, “MESSAGE”, che contiene la descrizione dell’incident, sarà di tipo Link. Infatti, nella figura 52, il campo “Messaggio” è evidenziato in blu e con un semplice click si verrà indirizzati ad un’altra pagina. Questa funzionalità verrà meglio chiarita nel paragrafo “Incident”.

Un altro aspetto da esaminare, com’è possibile vedere dagli screen del Tree Pane, sono i quattro buttons denominati “SPINNA EVENTO”, “NASCONDI EVENTO”, “APRI TICKET”, “COPIA”:

1. **SPIN**: è un button con template “Text with icon”, il cui “Behavior”, ossia l’azione generata quando il pulsante viene premuto, è la chiusura della pagina.

Le figure riportate di seguito, corrispondono, rispettivamente, alla scheda Processing del Tree Pane, contenente un processo *Spin*, ed al Property Editor contenente le informazioni relative a quel processo; il processo *Spin* è di tipo Execute Code, ossia nel momento in cui il pulsante viene premuto, verrà eseguito un blocco di codice PL/SQL riportato nella sezione Source. Il codice è così interpretabile: per rimuovere dagli eventi pinnati un incident, si deve modificare la tabella “incident” (UPDATE INCIDENT), ponendo il campo “pinned” uguale a 0 e non più ad 1, per l’evento che ha identificativo pari al valore del page item P7\_MM\_EVENT\_ID ed infine salvare le modifiche con l’istruzione COMMIT.

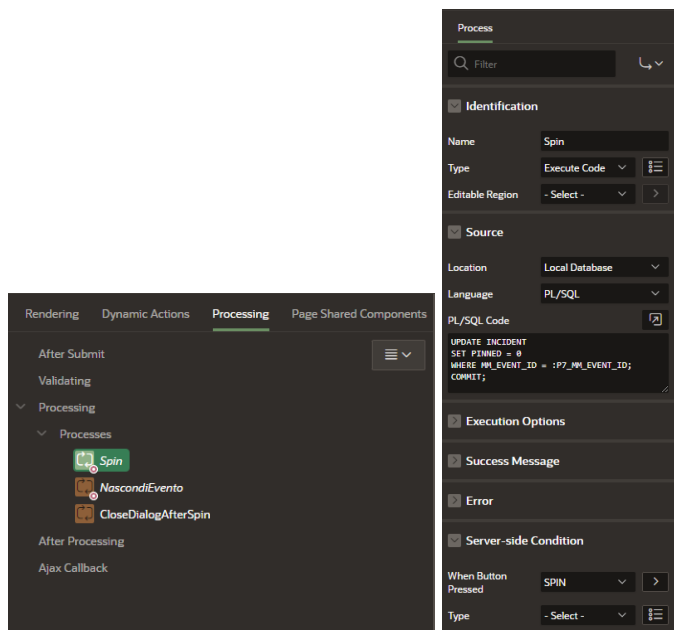


FIGURA 54: GESTIONE DEI PROCESSI LEGATI AL PULSANTE "SPIN" NEL TREE PANE E NEL PROPERTY EDITOR

2. **NASCONDIEVENTO**: come il precedente pulsante è di tipo Text with icon, ed ha un processo associato “*NascondiEvento*” che prevede l’esecuzione di codice PL/SQL alla pressione. Anche in questo caso verrà modificato un campo, hidden, settato a 1, all’interno della table incident, per l’evento che ha ID pari a P7\_MM\_EVENT\_ID.

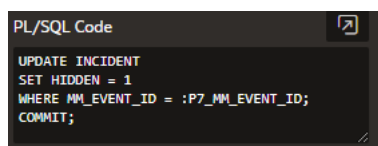


FIGURA 55: CODICE PL/SQL ASSOCIATO AL PROCESSO NASCONDI EVENTO

Nel momento in cui un evento viene nascosto non solo non verrà più mostrato nella timeline activity, ma non potrà più essere pinnato in quanto uno dei requisiti per essere posto in evidenza è che il campo hidden sia pari a 0.

Il processo *CloseDialogAfterSpin* è di tipo Close Dialog e consente di chiudere la finestra, in questo caso la modal dialog con la descrizione dell'evento ed i quattro pulsanti, non appena il processo *Spin* (se premuto il pulsante “spinna evento”) o il processo *NascondiEvento* (se premuto il rispettivo pulsante) saranno terminati.

3. APRITICKET: è un button la cui gestione non è stata ancora implementata e le sue funzionalità verranno meglio approfondite nel capitolo seguente.
4. COPIA: è un pulsante che alla pressione consente di copiare in forma testuale i dettagli dell'incident mostrato. Questa funzione viene gestita mediante una Dynamic Action *OnClickButtonCopia*, la quale dipende a sua volta da due azioni; un'azione di tipo Execute Server-Side Code che consente di eseguire del codice PL/SQL e un'altra di tipo Execute JavaScript Code.

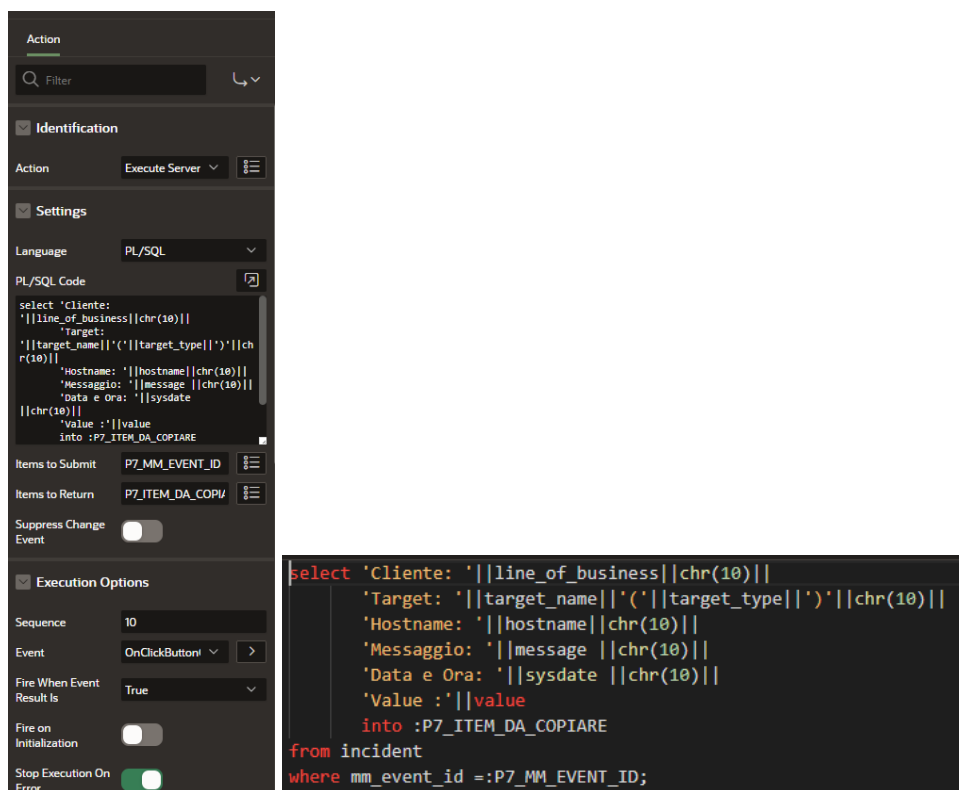


FIGURA 56: ACTION DI TIPO EXECUTE SERVER-SIDE CODE LEGATO AL PROCESSO ONCLICKBUTTONCOPIA

La prima azione, com'è visibile dalle figure sovrastanti, dipende dal blocco PL/SQL in cui i tutti i campi elencati nella clausola select, provenienti dalla tabella “incident” e con

event id pari a P7\_MM\_EVENT\_ID, vengono salvati nel page item P7\_ITEM\_DA\_COPIARE.

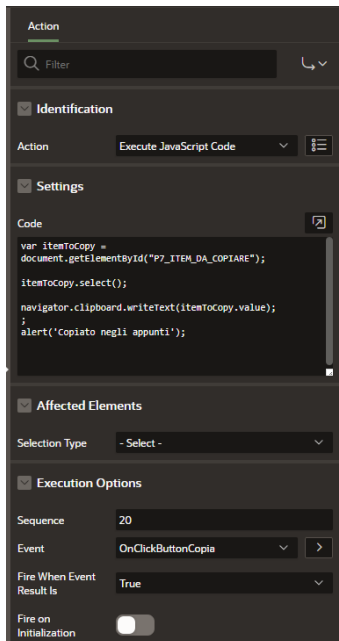


FIGURA 57: ACTION EXECUTE JAVASCRIPT CODE LEGATA AL PROCESSO ONCLICKBUTTONCOPIA

La seconda azione, Execute JavaScript Code, consente di definire ed eseguire un'azione che si avvale di codice JavaScript; la prima riga del codice si avvale della funzione *document.getElementById()* che ritorna un elemento, all'interno della variabile *itemToCopy*, che ha id specificato come argomento della funzione. In questo caso, P7\_ITEM\_DA\_COPIARE. L' *itemToCopy* verrà selezionato con funzione *select()* e di seguito, mediante la funzione *navigator.clipboard.writeText()*, che ha come argomento il valore della variabile *itemToCopy*, scrive la stringa di testo selezionata negli appunti di sistema (clipboard). Infine, se l'azione andrà a buon fine, comparirà un alert con scritto "Copiato negli appunti".

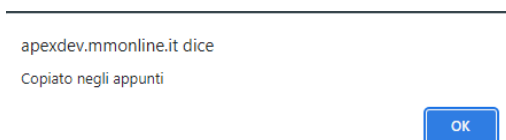


FIGURA 58: ALERT DI SUCCESSO PER L'AZIONE DI COPIA

Il risultato è riportato di seguito.

Cliente: Nome Cliente

Target: DNLAB\_blaix2(Database Instance)

Hostname: blaix2

Messaggio: 0 distinct types of ORA- errors have been found in the alert log.

Data e Ora: 2/16/2023

Value :1

Per concludere l'analisi della sezione Eventi Pinnati, è opportuno anche porre l'attenzione sulla Dynamic Action creata, DialogClosedAfterSpin, funzionale all'esecuzione di un'azione di Refresh della Timeline. Quando si verifica l'evento Dialog Closed, in questo caso la chiusura della finestra con i dettagli di un evento della Timeline a seguito della pressione di uno dei bottoni "Spinna Evento" o "Nascondi Evento", viene eseguita una TRUE Action Refresh che esegue l'aggiornamento degli Eventi Pinnati. Nel particolare, se un evento è stato soggetto a spin, la Timeline verrà aggiornata non presentando più quell'evento.

## 2.2.6 Ultimi Eventi

"Ultimi Eventi" è una region presente sotto Eventi Pinnati, gestita anch'essa come un Classic Report con template Activity Timeline, in quanto si propone di mostrare tutti gli incidenti rilevati nelle ultime due ore. Di seguito la query che consente di mostrare gli Ultimi eventi.

```
1  select message ,
2     mm_event_id USER_NAME,
3     to_char(updated,'HH24:MI') EVENT_DATE,
4     UPPER(substr(severity,1,2)) EVENT_TYPE,
5     (line_of_business||' - '||target_name||' - '|| target_type) EVENT_TITLE,
6     case
7         when severity='WARNING' then 'u-warning'
8         when severity='FATAL' then 'u-color-15'
9         when severity='CRITICAL' then 'u-danger'
10        when severity='CLEAR' then 'u-success'
11    end EVENT_STATUS,
12    substr(lifecycle_status,1,2) USER_AVATAR,
13    case
14        when lifecycle_status='MissionCritical' then 'u-color-15'
15        when lifecycle_status='Production' then 'u-color-24'
16        when lifecycle_status='Test' then 'u-color-35'
17        when lifecycle_status='Development' then 'u-color-35'
18    end USER_COLOR ,
19    apex_page.get_url(p_page => 7, p_items => 'P7_MM_EVENT_ID', p_values =>mm_event_id) EVENT_LINK,
20    message EVENT_DESC
21  from incident
22  where hidden = 0
23     and nvl(updated,created)>last_update_cc
24     and message not in (select *
25                          from all_notification_ignore
26                          where message like '%||message_pattern||%')
27     and created >= (select (max(created)-interval '2' hour)
28                    from incident
29                    where hidden =0
30                    and message not in( select *
31                                       from all_notification_ignore
32                                       where message like '%||message_pattern||%'))
33  order by mm_event_id;
34
35
```

FIGURA 59: QUERY CHE GESTISCE ULTIMI EVENTI

Gli elementi della Timeline, come USER\_NAME, EVENT\_DATE etc. si riferiscono agli stessi campi della table Incident come in Eventi Pinnati, mostrando le stesse informazioni; tuttavia,

si richiede di selezionare gli eventi che hanno data created superiore all'ultima modifica apportata last\_update\_cc, si desidera che il messaggio descrittivo dell>alert non sia contenuto nella tabella all\_notification\_ignore e che la detection dell'evento risalga alle ultime due ore.

## 2.3 Incident

L'applicazione AMS: Real-Time Incidents prevede la visualizzazione in forma tabellare di tutti gli allarmi ricevuti real-time; la pagina di riferimento è "Incident", raggiungibile non solo mediante il menù di navigazione, ma anche da alcune sezioni della page "Panoramica", aspetto, quest'ultimo, che verrà approfondito a seguito della presentazione delle funzioni e componenti della pagina "Incident". Accedendo alla sezione "Incident" dal menù di navigazione, la pagina renderizzata è la seguente:

Min Event Id	Cliente	Target	Tipo Target	Categoria	Messaggio
6251976	CAIPBON	CAIPBON2	Database Instance	Error	Metric Evaluation Error cleared.
6251977	AROOVAR		Database Instance	Jobs	Job RMAN_ARCHIVELOG_4 failed.
6251978	pdm		Database Instance	Jobs	Job RMAN_ARCHIVELOG_4 failed.
6251979	EMSP		Database Instance	Jobs	Job RMAN_ARCHIVELOG_4 failed.
6251980	qbase		Database Instance	Jobs	Job RMAN_ARCHIVELOG_4 failed.
6251981				Jobs	Event automatically cleared by event rule: Default, Clear job status change events older than 7 days.
6251982	kpromox6	kpromox61	Database Instance	Error	Metric Evaluation Error cleared.
6251983				Jobs	Event automatically cleared by event rule: Default, Clear job status change events older than 7 days.
6251984				Jobs	Event automatically cleared by event rule: Default, Clear job status change events older than 7 days.
6251985		ent1ondb	Database Instance	Error	Metric Evaluation Error cleared.
6251986	alfa1.aimag.priv		Host	Jobs	Job EXPDP_PMSDE2 is scheduled for execution.
6251987	alfa1.aimag.priv		Host	Jobs	Job EXPDP_PMSDE2 is running.
6251988	alfa0.aimag.priv		Host	Jobs	Job EXPDP_PMSDE2 is running.
6251989	alfa1.aimag.priv		Host	Jobs	Job EXPDP_PMSDE2 completed successfully.
6251990	dbprod1:3872		Agent	Availability	The current status of the target is Up.

FIGURA 60: PAGE INCIDENT

Com'è visibile, la pagina "Incident" mostra nel margine superiore l'indicatore di percorso (Breadcrumb), una barra di ricerca e dei campi posti immediatamente sotto; di seguito, una tabella con informazioni inerenti all'identificativo dell'incident, alle iniziali dell'etichetta dell'incident, evidenziate con differenti colori, al cliente, alla categoria a cui appartiene l'allarme, al target e alla sua tipologia e al messaggio descrittivo dell'allarme. La barra di ricerca ed i campi riportati sotto hanno lo scopo di permettere all'utente di filtrare i dati in tabella

selezionando dei valori precisi che i campi stessi possono assumere. Nella figura riportata di seguito, ad esempio, è stato selezionato il campo *lifecycle\_status* con valore “Mission Critical” per cui gli incident mostrati saranno solo taluni con etichetta “Mission Critical”.

Client	Instance	Target	Tipo Target	Categoria	Messaggio
6251985	ent1cnadb		Database Instance	Error	Metric Evaluation Error cleared.
6251986	alfa1.aimag.priv		Host	Jobs	Job EXPDP_PMSDE.2 is scheduled for execution.
6251987	alfa1.aimag.priv		Host	Jobs	Job EXPDP_PMSDE.2 is running.
6251988	alfa0.aimag.priv		Host	Jobs	Job EXPDP_PMSDE.2 is running.
6251989	alfa1.aimag.priv		Host	Jobs	Job EXPDP_PMSDE.2 completed successfully.

FIGURA 61: SMART FILTERS SUL REPORT INCIDENT

In fase di analisi dei requisiti, si è convenuto che gli “elementi” APEX più adeguati a realizzare le operazioni di filtraggio fossero gli Smart Filters, tipicamente composti da un filtro rappresentato da una barra di ricerca ed ulteriori filtri per selezionare il valore di un campo per cui si desidera filtrare i dati del report. Ciascun filtro associato ad un attributo degli incidents, include informazioni inerenti ai valori che quell’attributo può assumere ed il numero delle righe del report che possiedono quel valore. Passando all’ambiente di sviluppo, il page Designer avrà la seguente struttura:

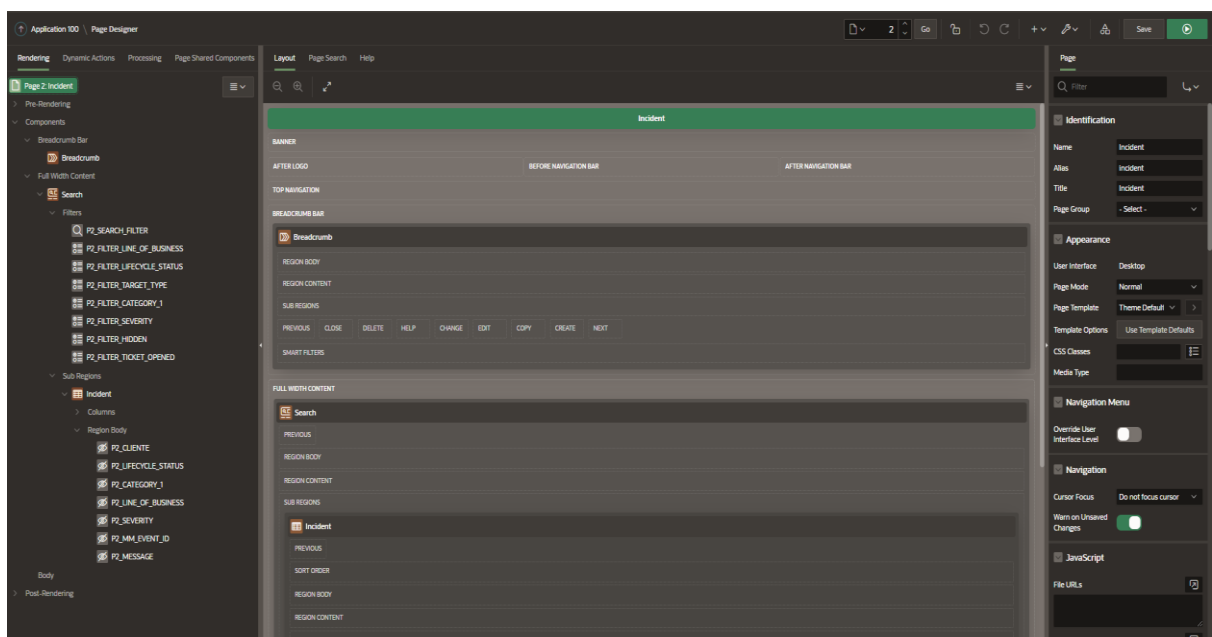


FIGURA 62: PAGE DESIGNER DELLA PAGINA INCIDENT

In fase di sviluppo, è stata creata una region denominata “Search”, com’è visibile dal Tree Pane di tipo Smart Filters. A seguire sono stati definiti tutti i filtri di due differenti tipologie, “Search” per *P2\_SEARCH\_FILTER* e “Radio Group” per i restanti *P2\_FILTER\_LINE\_OF\_BUSINESS*, *P2\_FILTER\_LIFECYCLE\_STATUS*, *P2\_FILTER\_TARGET\_TYPE*, *P2\_FILTER\_CATEGORY\_1*, *P2\_FILTER\_SEVERITY*, *P2\_FILTER\_HIDDEN*, *P2\_FILTER\_TICKET\_OPENED*; quest’ultimo non è stato implementato in quanto si riferisce ad una funzione dell’applicazione compresa tra gli sviluppi futuri. Gli screenshots mostrano il Property Editor del filtro di tipo “Search” e di uno dei filtri di tipo “Radio Group”. Il filtro “Search” rappresenta una barra di ricerca mentre il filtro “Radio Group” mostra valori multipli per un determinato attributo che l’utente può selezionare.

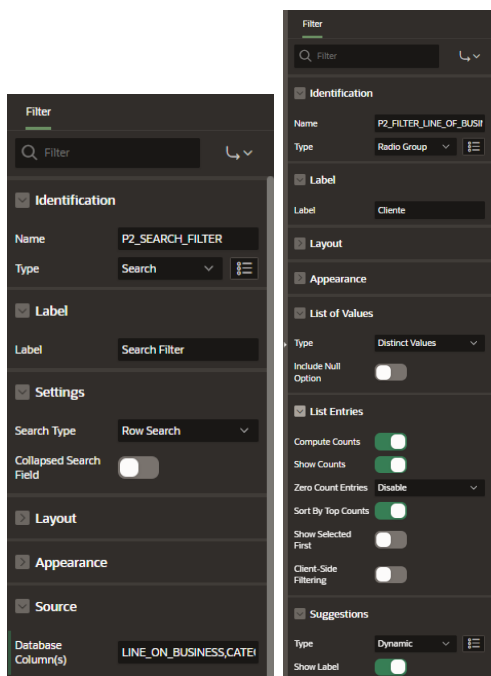


FIGURA 63: PROPERTY EDITOR DEI FILTRI DI TIPO SEARCH E RADIO GROUP

Proseguendo con gli elementi del Tree Pane, si incontra la region “Incident” di tipo Classic Report con un template standard. La query che gestisce il report intende restituire tutte le informazioni contenute nella tabella “Incident”, ma solo alcune di esse verranno mostrate; nel particolare, l’identificativo dell’allarme, il *target\_name* ed il *target\_type*, la *category*, il messaggio descrittivo e le iniziali della *severity* e *lifecycle\_status* entrambe gestite mediante lo statement *UPPER(substr(severity,1,2) sev* e *UPPER(substr(lifecycle\_status,1,2) life\_stat*. La funzione *UPPER()* è funzionale a tradurre la stringa in lettere maiuscole, mentre la funzione



*substr()* richiede come argomento un attributo e la posizione iniziale e finale della sottostringa che si vuole ottenere. L’istruzione “case” consente di attribuire un diverso colore alle sottostringhe *sev* e *life\_sat* in base ai loro valori utilizzando le classi di colore CSS come ‘u-danger’, ‘u-warning’ etc.

```

1  select MM_EVENT_ID,
2         INCIDENT_ID,
3         SEQUENCE_ID,
4         CLOUD_CONTROL_ID,
5         INCIDENT_STATUS,
6         LINE_OF_BUSINESS,
7         SEVERITY,
8         CATEGORY_1,
9         TARGET_NAME,
10        TARGET_TYPE,
11        HOSTNAME,
12        LIFECYCLE_STATUS,
13        EVENT_TYPE,
14        EVENT_NAME,
15        MESSAGE,
16        UPPER(substr(SEVERITY,1,2)) sev,
17        UPPER(substr(lifecycle_status,1,2)) life_stat,
18        VALUE,
19        CREATED,
20        UPDATED,
21        LAST_UPDATE_CC,
22        HIDDEN,
23        PINNED,
24        TICKET_OPENED,
25        SEVERITYPOINTS,
26        case UPPER(substr(SEVERITY,1,2))
27            when 'FA' then 'u-danger'
28            when 'CR' then 'u-hot'
29            when 'WA' then 'u-warning'
30            when 'CL' then 'u-color-20'
31            when 'IN' then 'u-color-16'
32        end sev_color_class,
33        case UPPER(substr(lifecycle_status,1,2))
34            when 'MI' then 'u-hot'
35            when 'PR' then 'u-danger'
36            when 'TE' then 'u-success'
37            when 'DE' then 'u-success'
38        end life_stat_color_class
39
40 from INCIDENT
41 where (decode(lifecycle_status,'Test','Development Test','Development','Development Test',lifecycle_status)=:P2_LIFECYCLE_STATUS OR :P2_LIFECYCLE_STATUS is null)
42 AND (CATEGORY_1 =:P2_CATEGORY_1 OR :P2_CATEGORY_1 is null)
43 AND (SEVERITY=:P2_SEVERITY OR :P2_SEVERITY is null)
44 AND (LINE_OF_BUSINESS=:P2_CLIENTE OR :P2_CLIENTE is null)

```

FIGURA 64: QUERY CHE GESTISCE IL REPORT INCIDENT

Inoltre, occorre focalizzare l’attenzione sulle istruzioni presenti nella where condition, in cui si impone di mostrare gli incident con valore di *lifecycle\_status* pari al page item *P2\_LIFECYCLE\_STATUS* o null, con valore di *category\_1* pari al page item *P2\_CATEGORY\_1*, oppure null, con valore di *severity* pari al page item *P2\_SEVERITY* ed infine con valore *line\_of\_business* pari a *P2\_CLIENTE* o null. La possibilità che i vari attributi abbiano valore pari al page item di riferimento, dipende infatti dalle funzionalità di alcuni componenti della page Panoramica. Infatti, se l’utente intende ispezionare un elemento, ad esempio Mission Critical Availability FATAL, di una delle Media List (Mission Critical, Production, Development Test), un semplice click sull’elemento reindirizza alla pagina Incident mostrando tutti gli eventi con etichetta Mission Critical, appartenenti alla categoria Availability e con severity FATAL. Lo stesso accade se nella PopupLOV dei Clienti ne viene selezionato uno, di conseguenza verranno mostrati solo gli incident di quei clienti. Di seguito viene riportato l’esempio appena descritto.

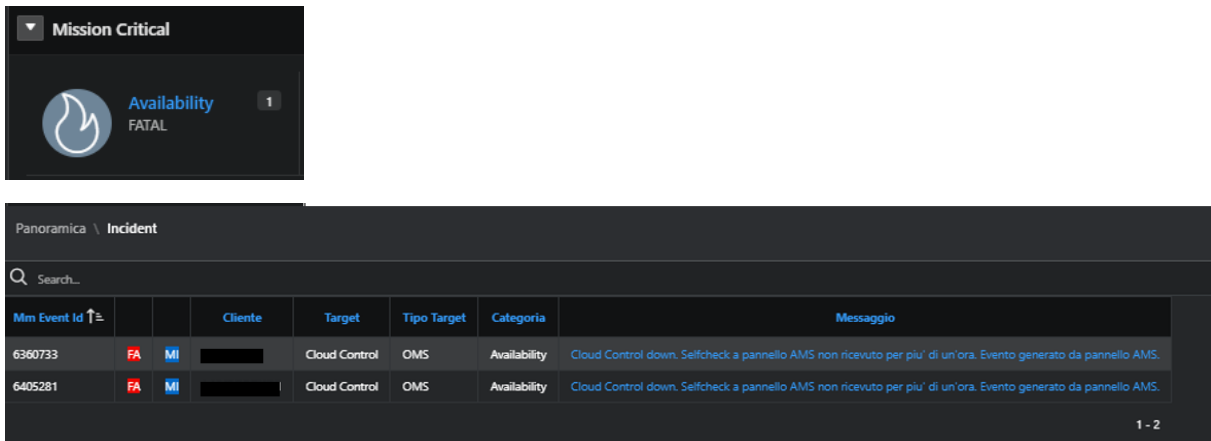


FIGURA 65: REPORT INCIDENT FILTRATO DALLA SELEZIONE DI MISSION CRITICAL, AVAILABILITY, FATAL

Un'ulteriore funzionalità da approfondire riguarda la colonna “Messaggio” del report Incident. Nel Page Designer la tipologia della colonna, visibile sia nel Tree Pane che nel Property Editor, è Link, vale a dire che compiendo un'azione di click sul messaggio descrittivo, evidenziato in blu, l'utente verrà indirizzato in un'altra pagina.

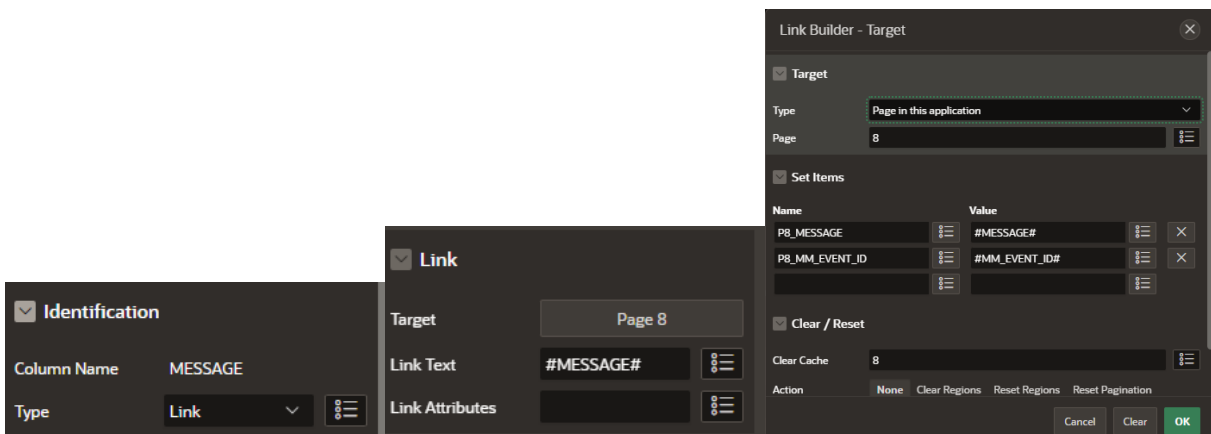


FIGURA 66: GESTIONE DELLA COLONNA MESSAGGIO DI TIPO LINK

Questa funzione viene gestita nell'attributo Link del Property Editor relativo alla colonna “Message”; accedendo al campo target, ciò che viene mostrato al developer è la finestra di dialogo coincidente con la terza immagine mostrata di sopra. Al developer è infatti richiesto di inserire la pagina target, ossia la pagina a cui l'utente viene indirizzato cliccando sul messaggio (in questo caso la 8). Inoltre, se risulta necessario ai fini delle logiche dell'applicazione, nella sezione Set Items, vengono riportati gli item (indicati con *Name*) della pagina target a cui associare valori di items o colonne della pagina corrente (indicati con *Value*). In questo caso Incident. Nel particolare, i page items di pagina 8, *P8\_MESSAGE* e *P8\_MM\_EVENT\_ID*,

acquisteranno rispettivamente i valori della colonna MESSAGE e della colonna MM\_EVENT\_ID (gli hashtag tra i nomi delle due colonne rappresentano il costrutto APEX per indicare il valore di una colonna). La pagina a cui l'utente viene indirizzato è 'Visualizza Incident' contenente una region di tipo Classic Report con template Value Attribute Pairs-Column, simile alla pagina "Evento Pinnato", raggiungibile cliccando su un evento nella Timeline degli "Eventi Pinnati" nella page "Panoramica".

The screenshot shows a web interface for viewing an incident. The title bar reads 'Panoramica \ Incident \ Visualizza Incident'. Below the title is a section titled 'Stato attuale Incident' with a 'Pin' button in the top right corner. The main content is a table with the following data:

Field	Value
Mm Event Id	6251977
Target Name	ARXWAR
Target Type	Database Instance
Lifecycle Status	MissionCritical
Severity	INFORMATIONAL
Category 1	Jobs
Hostname	ocmis-anti-01.ocmis.priv
Line Of Business	[REDACTED]
Event Name	JobStatus
Message	Job RMAN_ARCHIVELOG4 failed.
Value	

In the bottom right corner of the table area, there is a small indicator '1 - 1'.

**FIGURA 67: PAGINA VISUALIZZA INCIDENT**

Il report mostra le classiche informazioni relative ad un alert ed un pulsante "Pin", posto in alto a destra, funzionale ad esporlo tra gli Eventi Pinnati. Nel caso in cui l'evento in questione fosse già pinnato, il pulsante sarebbe identificato con il nome "Spin". Per implementare quanto detto, è stata creata la region di tipo Classic Report all'interno del Pane Tree nel Body della pagina, sono stati creati i due pulsanti, *Pin* e *Unpin*, ed inoltre i due page item *P8\_MM\_EVENT\_ID* e *P8\_MESSAGE*, descritti precedentemente. La scheda di rendering del Tree Pane è raffigurata di seguito.

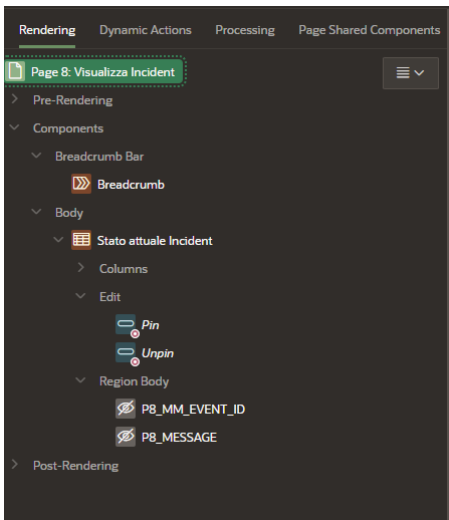


FIGURA 68: TREE PANE DELLA PAGINA VISUALIZZA INCIDENT

L'interrogazione SQL che ritorna i risultati mostrati nel report, seleziona tutte le colonne della tabella incident, di cui quelle che non sono oggetto di interesse verranno settate ad hidden nel Tree Pane, in cui il messaggio sia uguale al valore dell'item *P8\_MESSAGE* (*message= :P8\_MESSAGE*) e l'event id abbia valore del page item *P8\_MM\_EVENT\_ID* (*page item =:P8\_MM\_EVENT\_ID*).

```

1 select MM_EVENT_ID,
2        INCIDENT_ID,
3        SEQUENCE_ID,
4        CLOUD_CONTROL_ID,
5        INCIDENT_STATUS,
6        LINE_OF_BUSINESS,
7        CATEGORY_1,
8        TARGET_NAME,
9        TARGET_TYPE,
10       HOSTNAME,
11       LIFECYCLE_STATUS,
12       EVENT_TYPE,
13       EVENT_NAME,
14       MESSAGE,
15       SEVERITY,
16       VALUE,
17       CREATED,
18       UPDATED,
19       LAST_UPDATE_CC,
20       HIDDEN,
21       PINNED,
22       TICKET_OPENED,
23       SEVERITYPOINTS
24 from INCIDENT
25 where message =:P8_MESSAGE
26 and mm_event_id = :P8_MM_EVENT_ID

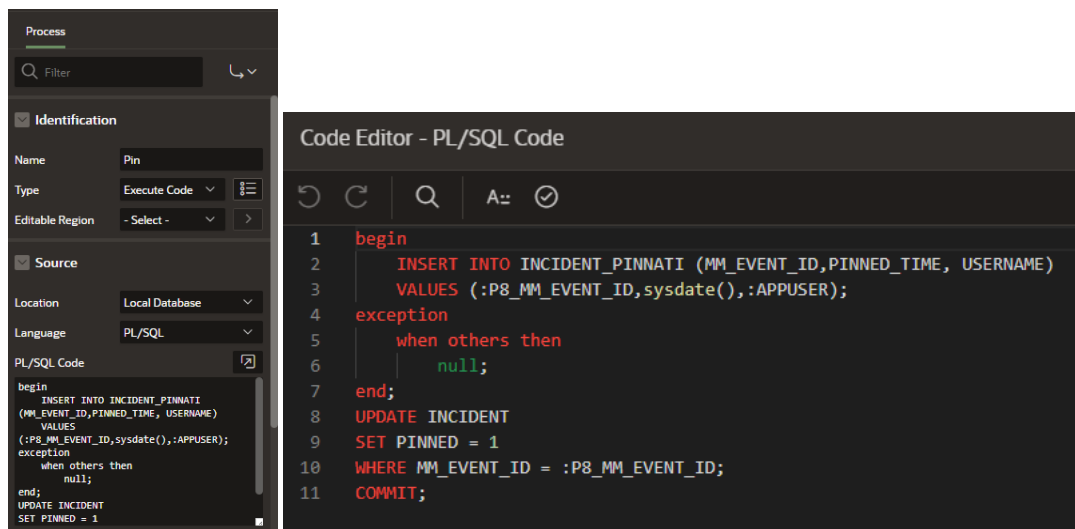
```

FIGURA 69: QUERY SQL PER VISUALIZZA INCIDENT

Il due pulsanti Pin e Unpin, sono entrambi due buttons di tipo “Text with Icon”, regolati rispettivamente da due processi omonimi:

- **Pin process:** è un processo di tipo Execute Code, che ha la funzione di eseguire il codice PL/SQL mostrato in figura. Alcune istruzioni da eseguire sono contenute nel blocco begin-exception-end e prevedono l'inserimento (INSERT INTO) nella table “INCIDENT\_PINNATI” con campi *MM\_EVENT\_ID*, *PINNED\_TIME* e *USERNAME*, che denotano l'id dell'evento, la data dell'azione di pin dell'evento e l'username dell'utente che ha eseguito il pin, con valori (*VALUES*) pari al valore del page item

*P8\_MM\_EVENT\_ID*, *sysdate()*, e il valore dell'item *APP\_USER*. L'exception viene utilizzata per gestire le eccezioni: se eseguendo le istruzioni nel corpo della begin si verifica un problema, mediante l'istruzione "exception when others then null", si impone di ignorare l'exception e continuare. Inoltre è richiesto di modificare (UPDATE) la table *incident*, impostando il valore di *pinned* a 1, facendo così apparire l'evento con id pari al page item *P8\_MM\_EVENT\_ID* nella Timeline Eventi Pinnati; il tutto viene reso permanente nel database con l'istruzione di *COMMIT*.



**FIGURA 70: PIN PROCESS**

**UnPin process:** è un processo, anch'esso di tipo *Execute Code*, che si attiva alla pressione dell' *UnPin* button. Ha la funzione di eseguire un blocco di codice *PL/SQL* che prevede di modificare la tabella "INCIDENT\_PINNATI", settando il valore dell'orario di unpin, pari all'orario attuale indicato dalla funzione *sysdate()*, per l'evento con id pari al page item *P8\_MM\_EVENT\_ID*; Unpin modifica inoltre la tabella *incident* settando il campo *pinned* dell'evento a 0, ad indicare che non sarà più nella timeline degli *Eventi Pinnati*; il blocco si conclude con l'istruzione di *COMMIT*.

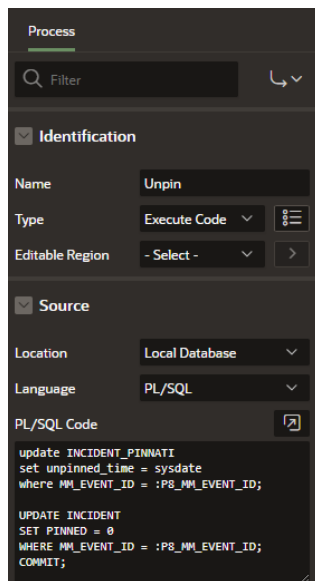


FIGURA 71: UNPIN PROCESS

## 2.4 Cloud Control

La pagina Cloud Control, accessibile all'utente sia dal menù di navigazione sia dall'elemento "Cloud Control OK", presente nella sezione "Altro" della pagina Panoramica, ha la funzione di mostrare lo stato del Cloud Control, anche detto Enterprise Manager, che opera l'anomaly detection sui sistemi dei clienti. Lo stato infatti può essere monitorato in base ad una serie di informazioni presentate nella pagina contenete una region di tipo Interactive Report, con campi inerenti al nome del cliente, "Ultimasync," alla data e orario dell'ultimo controllo avvenuto sul sistema del cliente, "Ultimoevento" ovvero alla data e orario dell'ultimo incident rilevato sul sistema del cliente, "Ultimoselfcheck" . Da ultimo, anche la data e l'orario dell'ultimo check per verificare che l'Enterprise Manager stia lavorando correttamente; in tal caso "Ultimostatus" conterrà il valore "Up", altrimenti il valore "Down".

La figura 71 mostra la pagina Cloud Control appena descritta e regolata nel Property Editor dalla query della figura 72.

The screenshot shows the 'Cloud Control' page in the AMS Mediamente application. It features a search bar, a 'Go' button, and an 'Actions' dropdown menu. The main content is a table with columns: 'Cliente', 'UltimaSync', 'UltimoEvento', 'UltimoStatus', and 'UltimoSelfCheck'. The table contains 20 rows of data, each representing a synchronization event for a client. The 'UltimoStatus' column shows 'Up' for all entries. The 'UltimoSelfCheck' column shows 'Up' for all entries. The 'UltimoEvento' column shows the date and time of the last event. The 'UltimaSync' column shows the date and time of the last sync. The 'Cliente' column is mostly empty, with a few entries visible.

Cliente	UltimaSync	UltimoEvento	UltimoStatus	UltimoSelfCheck
	26-JAN-2023 08:33AM	26-JAN-2023 07:00AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:39AM	26-JAN-2023 08:56AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:32AM	26-JAN-2023 08:54AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:39AM	26-JAN-2023 08:49AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:38AM	26-JAN-2023 08:04AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:34AM	26-JAN-2023 08:52AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:30AM	26-JAN-2023 08:47AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:34AM	26-JAN-2023 08:57AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:38AM	26-JAN-2023 08:55AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:25AM	26-JAN-2023 08:57AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:20AM	26-JAN-2023 08:57AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:32AM	26-JAN-2023 08:47AM	Up	26-JAN-2023 08:55AM
	30-SEP-2021 04:00PM	01-JUN-2021 11:59AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:32AM	26-JAN-2023 08:24AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:33AM	26-JAN-2023 08:53AM	Up	26-JAN-2023 08:55AM
	09-DEC-2021 12:53PM	09-DEC-2021 10:18AM	Up	09-DEC-2021 12:49PM
	26-JAN-2023 08:38AM	26-JAN-2023 06:22AM	Up	26-JAN-2023 08:55AM
	26-JAN-2023 08:36AM	26-JAN-2023 02:59AM	Up	26-JAN-2023 08:54AM
	26-JAN-2023 08:39AM	26-JAN-2023 08:40AM	Up	26-JAN-2023 08:55AM

FIGURA 72: PAGE CLOUD CONTROL

```

1  select cc.ID,
2         c.NOME as Cliente,
3         cc.ID_CLIENTE,
4         cc.API_TOKEN,
5         cc.TWILIO_ACCOUNTSID,
6         cc.TWILIO_APISID,
7         cc.CONFIG,
8         cc.LAST_HEARTBEAT as UltimaSync,
9         cc.LAST_SELFCHECK as UltimoSelfCheck,
10        case when cc.LAST_SELFCHECK_DOWNNUMBER=0 then 'Up'
11              when cc.LAST_SELFCHECK_DOWNNUMBER>0 THEN 'Down'
12        end UltimoStatus,
13        cc.LAST_EVENT as UltimoEvento,
14        cc.SCRIPT_VERSION,
15        cc.CALL_ON_DOWN,
16        cc.DISMISSED,
17        cc.HIDDEN_CC,
18        cc.VERSTON,
19        cc.VERSTON
20 FROM cloud_control cc, clienti c
21 WHERE dismissed=0 and cc.id_cliente=c.id

```

FIGURA 73: QUERY CLOUD CONTROL

## 2.5 Clienti

La pagina Clienti a cui si ha accesso diretto nel menu di navigazione, ha la finalità di mostrare la lista dei Clienti aziendali; selezionando uno di essi, l'utente sarà indirizzato ad un'altra pagina dell'applicazione con lo scopo di riassumere informazioni generali sul cliente, i relativi contatti, nonché i dati sull'infrastruttura dei sistemi dei clienti e una bacheca, in quattro differenti schede selezionabili. La pagina "Clienti" contiene una region di tipo classic report ed i clienti mostrati provengono direttamente dalla tabella CLIENTI. Com'è visibile nella figura x, il report mostra solo il nome del cliente, per cui tutte le altre colonne della tabella CLIENTI saranno impostate a "hidden" (nascosto), ad eccezione del campo NOME che è di tipo link. Infatti, la pagina target è la pagina 9 denominata "Cliente". Per visionare i dettagli relativi al cliente selezionato nella

nuova pagina “Cliente”, viene attribuito al page item P9\_ID\_CLIENTE il valore ID del cliente selezionato dalla pagina Clienti. Questi ultimi aspetti sono invece chiariti nella figura 74.

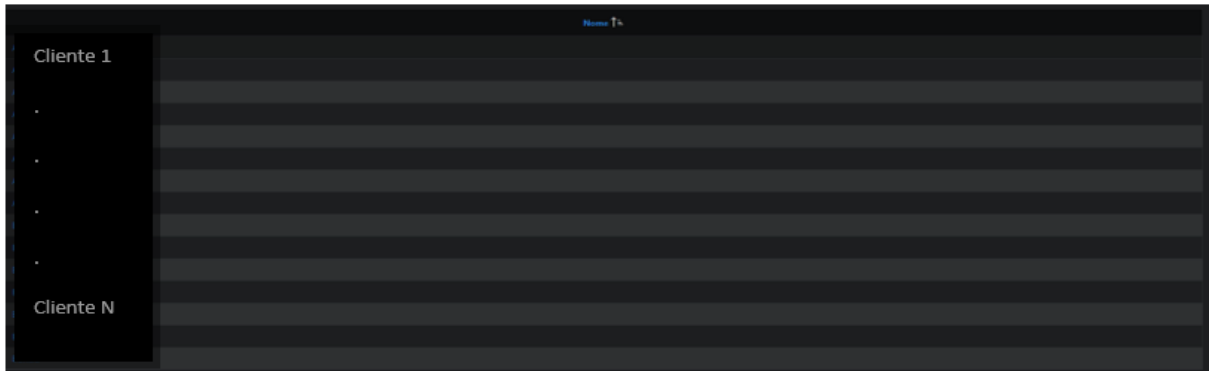


FIGURA 74: PAGINA CLIENTI

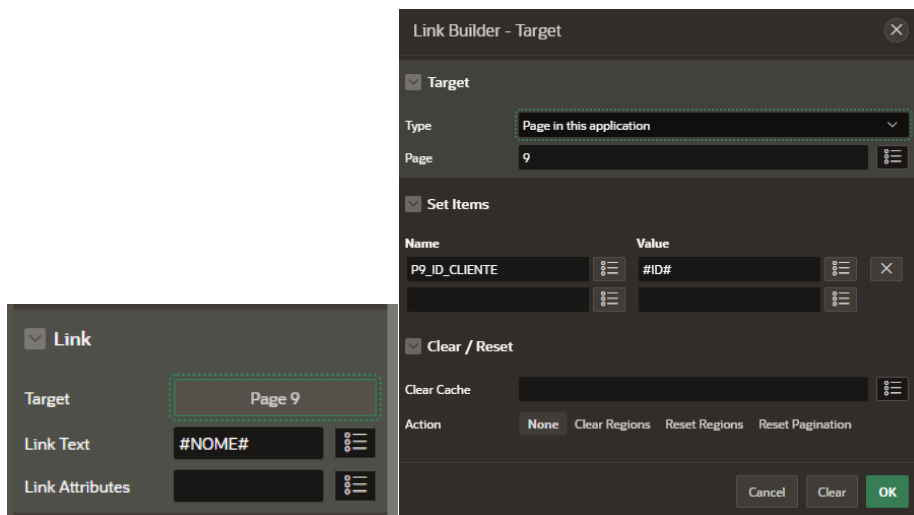


FIGURA 75: GESTIONE DELLA COLONNA “NOME” DI TIPO LINK

La pagina “Cliente” viene gestita mediante il componente APEX “Region Display Selector”, che consente di mostrare le regions della pagina come schede; in particolare, è presente una scheda in cui sono visibili tutte le sezioni semplicemente scorrendo la pagina verso il basso, mentre le altre schede, se selezionate, mostreranno solo la corrispondente region a cui sono collegate.



In questo caso è stata creata ed aggiunta nel Tree Pane della pagina Cliente, una region di tipo “Region Display Selector”. Nel body della pagina, sono state aggiunte quattro regions, corrispondenti alle schede della “Region Display Selector”:

- **Info Generali:** scheda di tipo classic report con template Value Attributes Pairs-Columns. La figura x mostra le informazioni generali del cliente selezionato, tra cui il referente aziendale e delle note sui vari incident gestiti.

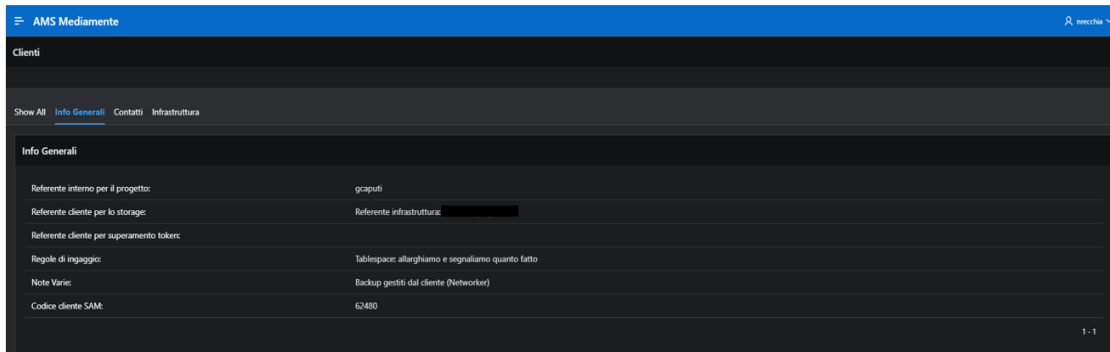


FIGURA 76: REGION DISPLAY SELECTOR CLIENTI

- **Contatti:** è una region di tipo Classic Report con template Value Attribute Pairs-Column, contenente una sub-region, anch’essa di tipo Classic Report con standard template, che funge da rubrica. Pertanto, dalla figura 76 , si può notare che vengono mostrate sia delle note sui contatti dei clienti, sia un report con numeri di telefono ed i relativi contatti, anche email. È possibile modificare una riga del report cliccando sul bottone presentato come icona con una matita, indirizzando direttamente l’utente ad una form. In alternativa, è possibile aggiungere un nuovo contatto mediante l’apposito bottone posto in alto a destra del report.

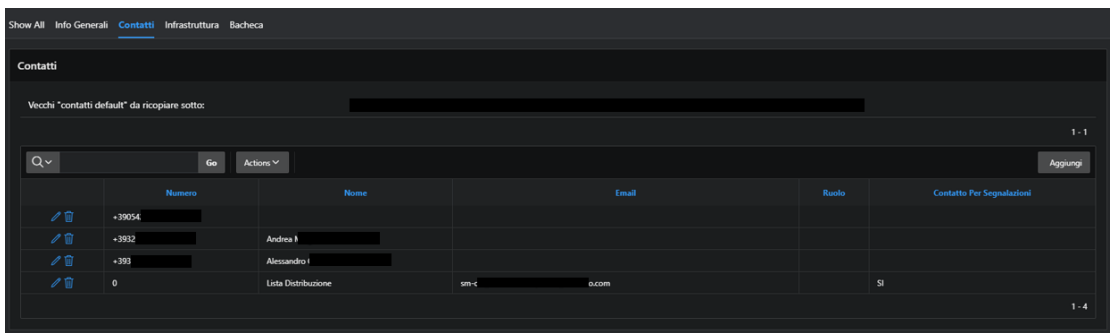


FIGURA 77: SEZIONE CONTATTI DEL REGION DISPLAY SELECTOR

- **Infrastruttura:** scheda divisa in due report di tipo Classic Report con template standard, entrambi mostrano le informazioni relative all'infrastruttura di due tipi di target dei sistemi del cliente.
- **Bacheca:** scheda la cui finalità è quella di mostrare avvisi rivolti ai clienti; essa non è stata ancora implementata.

# Capitolo 3

## Sviluppi Futuri

Nei successivi paragrafi verranno approfonditi aspetti relativi ai futuri sviluppi dell'applicazione AMS:Real-Time Incidents. Come anticipato nei precedenti capitoli, verranno meglio approfondite delle funzionalità da implementare in un secondo momento.

### 3.1 Mail Support

Nella Media List rappresentata dalla sezione “Altro” visibile nella pagina Panoramica, l'elemento “Mail support non lette” avrà lo scopo di indirizzare l'utente, mediante la semplice selezione, in una casella di posta dedicata interamente alla gestione degli incidents, in modo tale da distinguerla da una classica casella di posta aziendale.

### 3.2 Servizio Ticket

Un'altra funzionalità da rendere disponibile agli utenti è il servizio di ticketing; un ticket rappresenta un report in cui vengono mostrate informazioni relative ad un evento, come esso è stato gestito e da quale utente, eventuali ulteriori note d'interesse per il cliente etc. Il servizio di ticketing sarà disponibile da diversi punti dell'applicazione. Uno di essi sarà disponibile come elemento nella Media List “Altro” denominato “Ticket in stato aperto”: selezionandolo, l'utente verrà indirizzato ad un altro sistema, accessibile anche dai clienti, dove saranno visibili tutti i ticket creati e associati ad un determinato allarme.

Il servizio di “apertura” di un ticket sarà inoltre disponibile direttamente dalla region degli eventi pinnati presenti nella pagina Panoramica. Infatti, selezionando un evento in evidenza, si apre una finestra in modal dialog unitamente ad un pulsante “Apri Ticket” che verrà associato al sistema per l'attività di ticketing.

Anche nella pagina “Cliente”, accessibile selezionando un cliente direttamente dalla pagina “Clienti” presentata nel menù di navigazione, e nella pagina “Visualizza Incident”, discusse nel capitolo precedente, sarà presente un pulsante dedicato.

### **3.3 Invia Incident su Telegram**

Un altro servizio che verrà reso disponibile concerne l’invio di tutte le informazioni contenute nel report “Visualizza Incident” relativo ad un alert, mediante una chat dedicata sul social network Telegram. Questa funzionalità sarà gestita aggiungendo un pulsante “Invia su Telegram” direttamente nella pagina “Visualizza Incident”.

### **3.4 Bacheca**

La pagina “Bacheca” è già visibile nel menu di navigazione, tuttavia rappresenta una blank page che andrà a mostrare, una volta implementato, dei messaggi da segnalare ad un cliente. Il risultato sarà un report in cui è mostrato l’autore del messaggio, il cliente a cui è indirizzato, la data ed il messaggio per il cliente, laddove sarà, inoltre, possibile aggiungere gli avvisi mediante una form da compilare.

### **3.5 Calendario**

La pagina Calendario presente nel menu di navigazione, verrà implementata aggiungendo alla pagina una region di tipo Calendar, un componente messo a disposizione da APEX. Il calendario mostrerà i turni di lavoro degli utenti dell’applicazione AMS:Real-Time Incidents.

### **3.6 Centralino**

Questa sezione dell’applicazione, anch’essa attualmente visibile come una pagina vuota, ha lo scopo di fornire i nominativi e contatti dei dipendenti aziendali che utilizzano il sistema di

monitoraggio e sono reperibili per la ricezione di allarmi, la gestione degli stessi e chiamate dai clienti. Inoltre, sarà composta da due report, di cui uno mostra gli ultimi allarmi ricevuti e un altro le ultime chiamate ricevute da parte di un cliente con informazioni quali nome del cliente, numero del cliente da cui è stata effettuata la chiamata e la relativa data.

# Conclusioni

Il seguente elaborato di tesi si è proposto di illustrare il progetto di tirocinio, svolto presso l'azienda Mediamente Consulting, che ha condotto alla realizzazione di un'applicazione di monitoraggio in real-time di allarmi, suddivisi per grado di severità, provenienti dai sistemi enterprise dei clienti aziendali. Prima di descrivere nel dettaglio l'applicazione, si è ritenuto opportuno focalizzare l'attenzione sulla tecnologia utilizzata per l'implementazione: Oracle Application Express (APEX). In fase di analisi dei requisiti e progettazione, si è ritenuto che APEX fosse il development tool che meglio si prestasse al raggiungimento degli obiettivi prefissati. Tra questi vi era il proposito di ricercare il più idoneo data centric web application development tool per mostrare, mediante moderni strumenti di visualizzazione dati, gli alert rilevati e derivanti dai sistemi dei clienti, semplificando la gestione degli stessi da parte degli end users; inoltre, era necessario che lo strumento di sviluppo consentisse non solo la rappresentazione dei dati, ma anche lo storage, la modifica o qualsiasi attività di manipolazione di essi in real-time, il tutto con tempistiche di sviluppo ridotte. Oracle APEX è un development tool per enterprise web apps che si distingue dagli altri strumenti di business intelligence per la sua natura di piattaforma low code, consentendo di creare rapidamente applicazioni che richiedono decisamente minore stesura di codice e di risorse. Esso si presenta, inoltre, come uno strumento largamente diffuso per qualsiasi attività legata alla data visualization e data management, rendendo disponibili una molteplicità di tools per la reportistica dei dati e data utilities interni all'ambiente di sviluppo dedicati alla gestione dei dati.

Mediante Oracle APEX è stato possibile realizzare l'applicazione AMS:Real-Time Incidents composta da differenti pagine, ciascuna impiegata in diverse funzionalità implementate mediante gli strumenti reputati più adatti, quali Report con diversi template tra cui: Classic Report per la visualizzazione dei dati in forma tabellare e, nel caso di Interactive Report, sono stati inclusi strumenti per la personalizzazione dei dati, funzionalità di raggruppamento, esecuzione di calcoli, etc.; Media List e Activity Timeline per la visualizzazione e gestione dei dati, non attraverso la classica forma tabellare di reportistica, ma mediante elementi che rappresentano i dati e che contengono badges, icone, informazioni in evidenza, funzionalità per indirizzare l'utente verso altri punti dell'applicazione o sistemi/tecnologie esterni; Smart Filters utilizzati per filtrare e customizzare report in base alle informazioni di interesse contenute in un report; una barra di ricerca PopupLOV finalizzata a mostrare una lista di valori per una ricerca

mirata sui dati disponibili. Inoltre, mediante i numerosi strumenti per il development offerti nel Page Designer, l'applicazione presenta un alto livello di navigabilità tra pagine, mediante pulsanti gestiti con funzioni di linking. Pagine dell'applicazione sono state arricchite con pulsanti, che all'attivazione degli stessi, in quanto regolati da processi di diversa tipologia, generano delle specifiche azioni e comportano la modifica dei dati.

# Bibliografia

[1] <https://apex.oracle.com/it/>

[2] [https://apex.oracle.com/pls/apex/r/apex\\_pm/ut/getting-started](https://apex.oracle.com/pls/apex/r/apex_pm/ut/getting-started)

[3] <https://docs.oracle.com/en/database/oracle/apex/22.2/buildapplications.html>

[4] Oracle Application Express 3.2 – The Essentials and More, Arie Geller,  
Matthew Lyon

[5] Oracle APEX 20 For Beginners: A platform to develop stunning, scalable  
data-centric web apps fast, Riaz Ahmed