

UNIVERSITÀ DEGLI STUDI DI NAPOLI  
FEDERICO II



CORSO DI LAUREA TRIENNALE IN INGEGNERIA  
INFORMATICA

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE  
TECNOLOGIE DELL'INFORMAZIONE

**Basi di Dati**  
**Sistema Centralizzato di Prenotazioni**  
**Cinematografiche**

**Candidato:**

Vincenzo D'Angelo N46003125

**Professore:**

Vincenzo Moscato

Anno Accademico 2016/2017

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Specifiche dei requisiti</b>	<b>2</b>
<b>3</b>	<b>Configurazione del Database</b>	<b>4</b>
<b>4</b>	<b>Progettazione Concettuale</b>	<b>6</b>
4.1	Entità principali . . . . .	6
4.2	Entità di supporto . . . . .	6
4.3	Tabelle di Backup . . . . .	6
4.4	Relazioni principali . . . . .	7
<b>5</b>	<b>Progettazione Logica</b>	<b>9</b>
5.1	Tabelle principali . . . . .	9
5.2	Tabelle di Backup . . . . .	10
5.3	Vincoli e Integrità . . . . .	10
<b>6</b>	<b>Dimensionamento delle Tabelle</b>	<b>11</b>
<b>7</b>	<b>Implementazione SQL</b>	<b>14</b>
7.1	Utente Operativo . . . . .	14
7.1.1	Trigger: Aggiornamento posti disponibili . . . . .	15
7.1.2	Stored Procedure: Inserimento utente . . . . .	15
7.2	Utente di Backup . . . . .	16
7.2.1	Stored Procedure di backup . . . . .	16
<b>8</b>	<b>Esempi di Query SQL</b>	<b>18</b>
<b>9</b>	<b>Conclusioni e possibili estensioni</b>	<b>20</b>

# 1 Introduzione

Il presente elaborato descrive la progettazione e l'implementazione di un sistema informativo per la *gestione centralizzata delle prenotazioni cinematografiche* relative ai cinema distribuiti sul territorio svizzero.

Il progetto è stato sviluppato con l'obiettivo di integrare le basi di dati di diverse società locali — ciascuna responsabile di un singolo cantone — in un unico sistema centralizzato, capace di gestire in maniera uniforme informazioni su film, programmazioni, utenti e prenotazioni.

Il sistema è stato realizzato utilizzando Oracle Database [1] come DBMS, implementando un modello relazionale secondo standard SQL [2] e supportato da procedure e trigger in PL/SQL. La progettazione ha seguito metodologie classiche di modellazione concettuale e logica, con normalizzazione dei dati e definizione di vincoli di integrità. Sono stati inoltre previsti meccanismi di backup e gestione dei privilegi tramite ruoli, al fine di garantire sicurezza, affidabilità e consistenza dei dati.

## 2 Specifica dei requisiti

Nel contesto del progetto, si considera una multinazionale del settore informatico che ha recentemente acquisito diverse società locali responsabili dei servizi web per la prenotazione di biglietti presso cinema distribuiti sul territorio svizzero, ciascuna operante in un singolo cantone. L'obiettivo della società consiste nel fornire agli utenti un *servizio centralizzato* capace sia di consentire la consultazione della programmazione delle proiezioni dei film nei vari cinema, sia di permettere la prenotazione e l'acquisto online dei biglietti.

Il sistema deve gestire le seguenti informazioni principali:

- i **cinema**, con dati quali *nome, indirizzo, città, nazione, recapiti telefonici e numero di sale*;
- le **sale dei cinema**, identificate da *numero, superficie in metri quadrati, capienza e numerazione dei posti*;
- i **film in programmazione**, comprensivi di *titolo, recensione, regia, genere, anno, paese, durata, attori, data di uscita e distributore*;
- gli **utenti**, con *username, password, cognome, nome ed e-mail*;
- la **programmazione dei film**, specificando *giorno, orario e prezzo* dei biglietti;
- le **prenotazioni** e l'**acquisto dei biglietti** da parte degli utenti, comprensivi della *scelta dei posti*.

Le informazioni relative ai cinema di ciascun cantone, alla loro programmazione, agli orari e ai prezzi sono gestite dalle singole società locali, che nel tempo hanno sviluppato sistemi informativi indipendenti. Le informazioni sui film vengono invece recuperate da siti specializzati gestiti da terze parti, mentre i pagamenti con carta di credito devono essere eseguiti in modalità sicura tramite web service forniti da istituti di credito certificati.

Si prevede che, nell'arco del prossimo anno, la base dati dovrà gestire circa 50 cinema, 1000 utenti e 1000 film.

Per raggiungere tali obiettivi, il progetto prevede le seguenti attività principali:

1. la realizzazione della progettazione concettuale, logica e fisica della base di dati;
2. la creazione dell'intera base di dati mediante comandi DDL, specificando ove possibile i parametri di storage per ciascuna tabella;
3. l'implementazione in SQL di stored procedure, query, viste e trigger necessari per le operazioni richieste;
4. la definizione di indici opportuni per ottimizzare le prestazioni, l'individuazione della tipologia di controllo di concorrenza più adatta e la predisposizione di strategie di backup, recovery e replicazione, al fine di garantire l'affidabilità complessiva del sistema.

### 3 Configurazione del Database

Per la realizzazione del sistema informativo cinematografico sono stati creati due tablespace distinti, con l'obiettivo di separare i dati principali dai dati di backup. Il primo tablespace, denominato `ts_cinema`, è destinato a contenere tutte le informazioni operative del database, mentre il secondo, denominato `ts_backup`, è riservato esclusivamente alla memorizzazione dei backup. Entrambi i tablespace sono stati configurati con una dimensione iniziale di 50 MB e una capacità di autoestensione di 10 MB per volta, fino a un massimo di 12 GB, in modo da rispettare i limiti imposti dalla versione Oracle Database XE utilizzata.

```
CREATE TABLESPACE ts_cinema
DATAFILE 'C:\APP\VINCE\PRODUCT\21C\ORADATA\XE\ts_cinema.dbf'
SIZE 50M
AUTOEXTEND ON NEXT 10M MAXSIZE 12G;
```

Codice 1: Tablespace principale per i dati del database

```
CREATE TABLESPACE ts_backup
DATAFILE 'C:\APP\VINCE\PRODUCT\21C\ORADATA\XE\ts_backup.dbf'
SIZE 50M
AUTOEXTEND ON NEXT 10M MAXSIZE 12G;
```

Codice 2: Tablespace per backup

Successivamente, sono stati creati due utenti amministratori: il primo, `utente`, per la gestione ordinaria delle tabelle e delle operazioni sul database, e il secondo, `backup1`, dedicato alle operazioni di backup dei dati.

```
CREATE USER utente
IDENTIFIED BY 12345
DEFAULT TABLESPACE ts_cinema
TEMPORARY TABLESPACE TEMP;
```

Codice 3: Utente principale per operare sul database

```
CREATE USER backup1
IDENTIFIED BY 12345
DEFAULT TABLESPACE ts_backup
TEMPORARY TABLESPACE TEMP;
```

Codice 4: Utente per operazioni di backup

Per garantire le necessarie funzionalità di gestione, sono stati assegnati agli utenti i privilegi CONNECT, RESOURCE e UNLIMITED TABLESPACE. In particolare, utente può eseguire operazioni *CRUD* sulle tabelle principali, mentre backup1 gestisce esclusivamente i processi di backup.

```
GRANT CONNECT, RESOURCE, UNLIMITED TABLESPACE TO utente;
GRANT CONNECT, RESOURCE, UNLIMITED TABLESPACE TO backup1;
```

Codice 5: Assegnazione privilegi

La configurazione è stata verificata mediante query di controllo dei tablespace e degli utenti, confermando che i due tablespace erano presenti con i file fisici corretti e la dimensione prevista, e che entrambi gli utenti erano correttamente configurati nel database, associati ai rispettivi tablespace e con account attivi.

Questa organizzazione iniziale garantisce una separazione chiara tra dati operativi e dati di backup, assicurando al contempo una gestione sicura e controllata degli accessi al database.

## 4 Progettazione Concettuale

Il modello concettuale è stato sviluppato secondo la metodologia Entity–Relationship (E/R), individuando le principali entità e le relazioni tra di esse, con particolare attenzione all'integrità dei dati e alla gestione delle operazioni critiche.

### 4.1 Entità principali

- **CINEMA**: rappresenta ciascun cinema, con attributi relativi alla localizzazione, ai contatti e al numero di sale disponibili.
- **SALA**: associata a un cinema, identificata da numero, con informazioni su superficie e capienza.
- **FILM**: entità centrale del sistema, collegata a GENERE, PAESE, REGISTA e ATTORI.
- **PROIEZIONE**: lega un film a una sala in una determinata data e ora, includendo il prezzo e i posti disponibili.
- **UTENTE**: rappresenta gli utenti del sistema, con username, password, nome, cognome ed e-mail.
- **PRENOTAZIONE**: associa un utente a una proiezione, con informazioni su data e ora della prenotazione.
- **POSTO**: rappresenta i posti scelti dall'utente in fase di prenotazione.

### 4.2 Entità di supporto

- **GENERE, PAESE, REGISTA, ATTORI**: contengono le informazioni di riferimento per i film.
- **RECITA**: relazione tra ATTORI e FILM, che specifica il ruolo ricoperto da ciascun attore.

### 4.3 Tabelle di Backup

Per garantire la sicurezza e l'integrità dei dati, sono state previste tabelle di backup gestite dall'utente `backup1`. Queste tabelle replicano la struttura delle principali tabelle operative, permettendo di mantenere copie dei dati critici a fini di sicurezza e recupero. Le principali tabelle di backup sono:

- **Backup\_Prenotazioni**: memorizza le prenotazioni degli utenti.
- **Backup\_Proiezioni**: memorizza le proiezioni dei film nelle varie sale.
- **Backup\_Posto**: memorizza i posti prenotati dagli utenti per ciascuna proiezione.

Queste tabelle hanno le stesse chiavi primarie e chiavi esterne delle tabelle operative corrispondenti, garantendo la coerenza dei dati replicati senza introdurre nuove relazioni logiche nel modello principale.

#### 4.4 Relazioni principali

- Un **CINEMA** può contenere più **SALE**.
- Una **SALA** può ospitare più **PROIEZIONI**.
- Un **FILM** può essere proiettato in più **PROIEZIONI**.
- Un **UTENTE** può effettuare più **PRENOTAZIONI**.
- Ogni **PRENOTAZIONE** può includere più **POSTI**.
- La relazione **RECITA** collega più **ATTORI** a ciascun **FILM** con il ruolo specifico ricoperto.

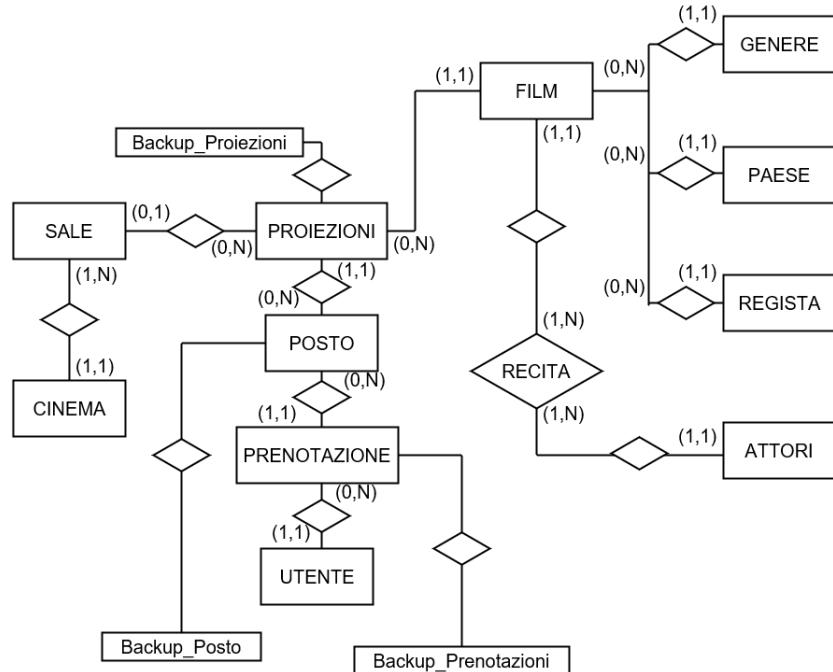


Figura 1: Modello E/R

Il modello ER così definito consente di rappresentare in maniera completa le informazioni relative a cinema, sale, film, utenti, proiezioni e prenotazioni, integrando al contempo le tabelle di backup per garantire la sicurezza e l'affidabilità.

dei dati. La progettazione è stata pensata per supportare le operazioni quotidiane del sistema, la gestione delle prenotazioni e la possibilità di recupero dati in caso di malfunzionamenti o errori.

## 5 Progettazione Logica

Il modello logico relazionale è stato derivato dal modello concettuale E/R, normalizzato fino alla terza forma normale (3NF) per eliminare ridondanze e garantire integrità dei dati.

### 5.1 Tabelle principali

- **UTENTE(Username, Password, Nome, Cognome, Email)**  
Rappresenta gli utenti del sistema. Gli attributi Password, Nome e Cognome sono obbligatori.
- **CINEMA(IDCinema, Nome, Indirizzo, Citta, Cantone, Telefono, Num\_Sale, Nazione)**  
Contiene le informazioni sui cinema. Ogni cinema può avere più sale associate.
- **SALE(Numero, CodCinema : CINEMA, Capienza, Superficie)**  
Rappresenta le singole sale all'interno dei cinema.
- **GENERE(IDGenere, Genere)**  
Rappresenta il genere cinematografico dei film.
- **PAESE(IDPaese, NomePaese)**  
Contiene i paesi di produzione dei film.
- **REGISTA(IDRegista, Nome, Cognome)**  
Contiene i registi dei film.
- **ATTORI(IDAttore, Nome, Cognome, DDN, Nazionalita)**  
Contiene gli attori dei film.
- **FILM(IDFilm, CodGenere : GENERE, CodPaese : PAESE, CodRegista : REGISTA, Titolo, Recensione, Anno, Durata, Data\_Uscita, Distributore)**  
Contiene le informazioni sui film.
- **RECITA(IDRecita, CodFilm : FILM, CodAttore : ATTORI, Ruolo)**  
Associa gli attori ai film con il ruolo specifico.
- **PROIEZIONI(IDProiezioni, CodSala : SALE, CodFilm : FILM, Giorno, Ora, Prezzo, Posti\_Dispo)**  
Contiene le informazioni relative alle proiezioni dei film nelle varie sale.
- **PRENOTAZIONI(IDPren, UsernameUtente : UTENTE, Data, Ora)**  
Rappresenta le prenotazioni effettuate dagli utenti.

- **POSTO(CodProiezioni, Numero, CodPren, CodPren : PRENOTAZIONI, CodProiezioni : PROIEZIONI)**

Rappresenta i posti selezionati dagli utenti per ciascuna prenotazione.

## 5.2 Tabelle di Backup

Per garantire la sicurezza dei dati, sono state create tabelle di backup replicate dall'utente `backup1`. Le tabelle di backup hanno la stessa struttura delle tabelle operative principali:

- **Backup\_Pren**: replica di PRENOTAZIONI
- **Backup\_Proiezioni**: replica di PROIEZIONI
- **Backup\_Posto**: replica di POSTO

Queste tabelle consentono di mantenere copie dei dati critici senza introdurre nuove relazioni logiche, supportando strategie di backup e recupero dati.

## 5.3 Vincoli e Integrità

Per tutte le tabelle principali e di backup sono stati definiti:

- Vincoli di chiave primaria e chiave esterna per garantire l'integrità referenziale;
- Vincoli NOT NULL su attributi obbligatori;
- Vincoli UNIQUE dove necessario;
- Domini coerenti con il tipo di dato e la lunghezza massima degli attributi.

Questo modello logico garantisce coerenza, riduzione delle ridondanze e facilità nella gestione delle operazioni di inserimento, aggiornamento, cancellazione e backup dei dati.

## 6 Dimensionamento delle Tabelle

Si suppone che, nell'arco del prossimo anno, la base di dati dovrà gestire le seguenti quantità di informazioni:

- circa 50 cinema;
- circa 1000 utenti;
- circa 1000 film;
- ogni cinema dispone di circa 10 sale;
- ogni film ha 1 regista e 10 attori;
- ogni prenotazione può comprendere fino a 200 posti;
- circa 5 milioni di proiezioni complessive.

Per il calcolo dello spazio occupato dai dati si sono adottate le seguenti convenzioni:

- NUMBER: 4 byte per ogni valore;
- CHAR (n) e VARCHAR2 (n) : 1 byte per carattere;
- DATE: 7 byte;
- TIMESTAMP: 7 byte;
- REAL: 4 byte.

Tabella	Attributo	Tipo e dimensione	Totale stimato (Byte)
UTENTE (1000)	Username Password Nome Cognome Email	VARCHAR2(20) VARCHAR2(20) VARCHAR2(30) VARCHAR2(30) VARCHAR2(50)	20K 20K 30K 30K 50K
<b>Totale UTENTE</b>			150K
CINEMA (50)	IDCinema Nome Indirizzo Città Cantone Telefono NumSale Nazione	VARCHAR2(10) VARCHAR2(50) VARCHAR2(100) VARCHAR2(30) VARCHAR2(30) VARCHAR2(15) NUMBER VARCHAR2(30)	500B 2.5KB 5KB 1.5KB 1.5KB 750B 200B 1.5KB
<b>Totale CINEMA</b>			circa 13.45KB

Tabella	Attributo	Tipo e dimensione	Totale stimato (Byte)
SALE (500)	Numero CodCinema Capienza Superficie	VARCHAR2(10) VARCHAR2(10) NUMBER NUMBER	5KB 5KB 2KB 2KB
<b>Totale SALE</b>			circa 14KB
GENERE (50)	IDGenere Genere	VARCHAR2(10) VARCHAR2(30)	500B 1.5KB
<b>Totale GENERE</b>			circa 2KB
PAESE (100)	IDPaese NomePaese	VARCHAR2(10) VARCHAR2(30)	1KB 3KB
<b>Totale PAESE</b>			4KB
REGISTA (1000)	IDRegista Nome Cognome	VARCHAR2(10) VARCHAR2(30) VARCHAR2(30)	10KB 30KB 30KB
<b>Totale REGISTA</b>			70KB
ATTORI (1000)	IDAttore Nome Cognome DDN Nazionalita	VARCHAR2(10) VARCHAR2(30) VARCHAR2(30) DATE VARCHAR2(30)	10KB 30KB 30KB 7KB 30KB
<b>Totale ATTORI</b>			107KB
FILM (1000)	IDFilm Titolo Recensione Anno Durata Data <sub>Uscita</sub> Distributore CodGenere CodPaese CodRegista	VARCHAR2(10) VARCHAR2(100) VARCHAR2(400) NUMBER NUMBER DATE VARCHAR2(50) VARCHAR2(10) VARCHAR2(10) VARCHAR2(10)	10KB 100KB 400KB 4KB 4KB 7KB 50KB 10KB 10KB 10KB
<b>Totale FILM</b>			circa 605KB
RECITA (10000)	IDRecita Ruolo CodFilm CodAttore	VARCHAR2(10) VARCHAR2(30) VARCHAR2(10) VARCHAR2(10)	100KB 300KB 100KB 100KB
<b>Totale RECITA</b>			circa 600KB
PROIEZIONI (5000)	IDProiezione CodSala CodCinema CodFilm Giorno Ora Prezzo Posti_Dispo	VARCHAR2(10) VARCHAR2(10) VARCHAR2(10) VARCHAR2(10) DATE TIMESTAMP NUMBER NUMBER	50KB 50KB 50KB 50KB 35KB 35KB 20KB 20KB
<b>Totale PROIEZIONI</b>			circa 310KB

Tabella	Attributo	Tipo e dimensione	Totale stimato (Byte)
PRENOTAZIONI (5 milioni)	IDPren DataPren OraPren UsernameUtente	VARCHAR2(10) DATE TIMESTAMP VARCHAR2(20)	50MB 35MB 35MB 100MB
<b>Totale PRENOTAZIONI</b>			circa 220MB
POSTO (1 miliardo)	Numero CodProiezione CodPren	NUMBER VARCHAR2(10) VARCHAR2(10)	4GB 10GB 10GB
<b>Totale POSTO</b>			circa 24GB
Backup (PROIEZIONI, POSTO, PRENOTAZIONI)	–	–	circa 24GB + 220MB + 310KB

Dal calcolo dello spazio totale emerge chiaramente che le tabelle di tipo transazionale come POSTO e PRENOTAZIONI rappresentano i componenti principali del database in termini di occupazione su disco, mentre le tabelle anagrafiche e di dominio operativo hanno un peso trascurabile rispetto ai dati operativi.

I dati di backup replicano le tabelle operative, permettendo di gestire con sicurezza la memorizzazione storica delle prenotazioni e delle proiezioni.

Sulla base delle stime riportate nella tabella di dimensionamento, il **tablespace principale** (TS\_CINEMA) dovrà contenere circa **25 GB**, mentre il **tablespace di backup** (TS\_BACKUP) dovrà gestire una quantità di dati simile, stimata in circa 25 GB + 220 MB + 310 KB, comprendendo le copie delle tabelle POSTO, PRENOTAZIONI e PROIEZIONI. Questa configurazione garantisce una gestione sicura dei dati storici e operativi, assicurando la disponibilità e l'integrità delle informazioni anche in caso di necessità di recupero.

## 7 Implementazione SQL

Per la realizzazione del database *Cinema*, sono stati utilizzati due script distinti, ciascuno eseguito con un utente specifico: uno per l'utente operativo principale e uno per l'utente destinato alla gestione dei backup. Questa separazione permette di isolare le operazioni operative dalle procedure di archiviazione e di garantire una gestione sicura dei dati storici.

### 7.1 Utente Operativo

Lo script destinato all'utente operativo (`utente`) definisce lo schema principale del database. In particolare, vengono create le tabelle anagrafiche, le tabelle operative, gli indici, i trigger e le stored procedure necessarie per la gestione delle prenotazioni e delle proiezioni.

**Tabella FILM**

```
CREATE TABLE FILM (
    IDFilm      VARCHAR2(10) PRIMARY KEY,
    Titolo       VARCHAR2(100) NOT NULL,
    Recensione   VARCHAR2(400),
    Anno         NUMBER,
    Durata       NUMBER,
    Data_Uscita  DATE,
    Distributore VARCHAR2(50),
    CodGenere    VARCHAR2(10),
    CodPaese     VARCHAR2(10),
    CodRegista   VARCHAR2(10),
    FOREIGN KEY (CodGenere) REFERENCES GENERE(IDGenere),
    FOREIGN KEY (CodPaese) REFERENCES PAESE(IDPaese),
    FOREIGN KEY (CodRegista) REFERENCES REGISTA(IDRegista)
);
```

Codice 6: Creazione tabella FILM

### 7.1.1 Trigger: Aggiornamento posti disponibili

Il trigger `trg_posti_disponibili` viene eseguito automaticamente dopo l'inserimento di un nuovo posto nella tabella POSTO. Lo scopo è aggiornare il numero di posti disponibili nella proiezione corrispondente.

```
CREATE OR REPLACE TRIGGER trg_posti_disponibili
AFTER INSERT ON POSTO
FOR EACH ROW
BEGIN
    -- Riduce di uno il campo Posti_Dispo nella tabella PROIEZIONI
    -- riferito alla proiezione appena occupata
    UPDATE PROIEZIONI
    SET Posti_Dispo = Posti_Dispo - 1
    WHERE IDProiezione = :NEW.CodProiezione;
END;
```

Codice 7: trigger `trg_posti_disponibili`

### 7.1.2 Stored Procedure: Inserimento utente

La procedura `InsertUtente` permette di inserire un nuovo utente nello schema UTENTE. Viene effettuato un controllo preventivo per verificare che lo username non sia già presente, in caso contrario viene loggato un messaggio nella tabella Log.

```
CREATE OR REPLACE PROCEDURE InsertUtente (
    p_Username IN UTENTE.Username%TYPE,
    p_Password IN UTENTE.Password%TYPE,
    p_Nome     IN UTENTE.Nome%TYPE,
    p_Cognome  IN UTENTE.Cognome%TYPE,
    p_Email    IN UTENTE.Email%TYPE
)
AS
    v_count INTEGER;
BEGIN
    -- Controllo dell'esistenza dello username
    SELECT COUNT(*) INTO v_count FROM UTENTE WHERE Username = p_Username
    ;
    IF v_count > 0 THEN
        -- Inserisce un log in caso di tentativo di registrazione
        -- duplicata
        INSERT INTO Log (Messaggio)
        VALUES ('Tentativo di registrazione con username già esistente: '
        || p_Username);
    ELSE
        -- Inserimento del nuovo utente
        INSERT INTO UTENTE (Username, Password, Nome, Cognome, Email)
        VALUES (p_Username, p_Password, p_Nome, p_Cognome, p_Email);
        COMMIT;
    END IF;
END;
```

Codice 8: procedura `InsertUtente`

## Grant a backup1

I grant consentono all'utente backup1 di leggere, inserire e cancellare i dati delle tabelle operative, in modo da eseguire correttamente le procedure di backup senza avere privilegi di amministratore completo

```
GRANT SELECT, INSERT, DELETE ON PROIEZIONI TO backup1;
GRANT SELECT, INSERT, DELETE ON POSTO TO backup1;
GRANT SELECT, INSERT, DELETE ON PRENOTAZIONI TO backup1;
```

Codice 9: grant a backup1

## 7.2 Utente di Backup

Lo script per l'utente di backup (backup1) consente la creazione delle tabelle di archiviazione, corrispondenti alle principali tabelle operative (PROIEZIONI, POSTO, PRENOTAZIONI), senza popolarle inizialmente.

### Backup tabella PROIEZIONI

```
CREATE TABLE Backup_Proiezioni AS
SELECT * FROM UTENTE.PROIEZIONI WHERE 1=0;
```

Codice 10: Creazione tabella di backup per PROIEZIONI

### 7.2.1 Stored Procedure di backup

La gestione dei backup viene affidata ad una procedura automatica che copia i dati storici e li rimuove dallo schema principale.

```
CREATE OR REPLACE PROCEDURE BACKUP_PROIEZIONI_STORICHE AS
BEGIN
    -- Copia proiezioni vecchie
    INSERT INTO BACKUP_PROIEZIONI
    SELECT * FROM UTENTE.PROIEZIONI
    WHERE GIORNO < SYSDATE - 30;

    -- Copia i posti associati alle proiezioni vecchie
    INSERT INTO BACKUP_POSTO
    SELECT P. *
    FROM UTENTE.POSTO P
    WHERE P.CODPROIEZIONE IN (
        SELECT IDPROIEZIONE
        FROM UTENTE.PROIEZIONI
        WHERE GIORNO < SYSDATE - 30
    );

    -- Copia le prenotazioni collegate ai posti
    INSERT INTO BACKUP_PREN
    SELECT PR. *
    FROM UTENTE.PRENOTAZIONI PR
    WHERE PR.IDPREN IN (
```

```

SELECT CODPREN
FROM UTENTE.POSTO
WHERE CODPROIEZIONE IN (
    SELECT IDPROIEZIONE
    FROM UTENTE.PROIEZIONI
    WHERE GIORNO < SYSDATE - 30
)
);

-- Cancella i record vecchi dallo schema UTENTE
DELETE FROM UTENTE.POSTO WHERE CODPROIEZIONE IN (
    SELECT IDPROIEZIONE
    FROM UTENTE.PROIEZIONI
    WHERE GIORNO < SYSDATE - 30
)
;

DELETE FROM UTENTE.PRENOTAZIONI WHERE IDPREN IN (
    SELECT CODPREN
    FROM UTENTE.POSTO
    WHERE CODPROIEZIONE IN (
        SELECT IDPROIEZIONE
        FROM UTENTE.PROIEZIONI
        WHERE GIORNO < SYSDATE - 30
    )
)
;

DELETE FROM UTENTE.PROIEZIONI WHERE GIORNO < SYSDATE - 30;
COMMIT;
END;

```

Codice 11: Stored Procedure di backup

Questa organizzazione consente di mantenere separati i dati attivi da quelli storici, garantendo sia la sicurezza sia la facilità di manutenzione del database.

## 8 Esempi di Query SQL

In questa sezione vengono illustrati alcuni esempi di query eseguite sul database CINEMA, utili per interrogazioni tipiche come la ricerca di film, la verifica delle prenotazioni o il conteggio di record per categoria. Le query sono state eseguite tramite l'utente principale utente.

### Film prodotti in un determinato Paese

```
SELECT F.Titolo
FROM FILM F
JOIN PAESE P ON F.CodPaese = P.IDPaese
WHERE P.NomePaese = 'Italia';
```

Il risultato di questa query restituisce i titoli dei film prodotti in Italia, consentendo di filtrare il catalogo per paese di produzione.

### Titoli e generi dei film recenti

```
SELECT F.Titolo, G.Genere
FROM FILM F
JOIN GENERE G ON F.CodGenere = G.IDGenere
WHERE F.Data_Uscita > TO_DATE('01/01/2001', 'DD/MM/YYYY');
```

Questa interrogazione permette di ottenere i film usciti dopo il 2001, accompagnati dal rispettivo genere, utile per analisi temporali o classificazioni tematiche.

### Film diretti da un regista specifico

```
SELECT F.Titolo
FROM FILM F
JOIN REGISTA R ON F.CodRegista = R.IDRegista
WHERE R.Nome = 'Ridley' AND R.Cognome = 'Scott';
```

Il comando sopra elenca tutti i film diretti da Ridley Scott, fornendo uno strumento rapido per consultare le opere di un regista specifico.

### Conteggio dei film per nazione

```
SELECT P.NomePaese, COUNT(F.IDFilm) AS NumeroFilm
FROM PAESE P
JOIN FILM F ON F.CodPaese = P.IDPaese
GROUP BY P.NomePaese;
```

Questa query restituisce il numero di film prodotti per ciascun paese, permettendo di analizzare la distribuzione geografica della produzione cinematografica.

## Film senza recensione

```
SELECT F.Titolo, G.Genere
FROM FILM F
JOIN GENERE G ON F.CodGenere = G.IDGenere
WHERE F.Recensione IS NULL;
```

Con questa interrogazione si individuano i film privi di recensione, utile per attività di aggiornamento dei dati o per analisi sulla completezza delle informazioni disponibili.

Questi esempi mostrano come interrogare il database per ottenere informazioni sui film, generi e registi, utile sia per operazioni di analisi che per il controllo dei dati inseriti.

## 9 Conclusioni e possibili estensioni

Il progetto ha permesso di applicare in maniera completa le competenze acquisite nell'ambito della progettazione e gestione di basi di dati relazionali, integrando aspetti di modellazione concettuale, logica e fisica con componenti di programmazione PL/SQL.

La soluzione proposta si è dimostrata:

- Affidabile, grazie alla presenza di procedure di backup e meccanismi di recovery;
- Sicura, attraverso una gestione granulare dei ruoli e dei privilegi;
- Realisticamente applicabile in contesti aziendali o universitari.

Tra le possibili estensioni future si prevedono:

- Piena integrazione con servizi esterni e sistemi di terze parti;
- Automazione dei processi di aggiornamento e manutenzione dei dati;
- Sviluppo di un portale web per consentire l'interazione diretta con gli utenti finali.

## Riferimenti

- [1] Oracle Corporation. *Oracle Database Concepts*. Accessed: 2025-10-26. 2025.  
URL: <https://docs.oracle.com/en/database/>.
- [2] *Information technology — Database languages — SQL*. ISO/IEC, 2016.