

Tesi Base di Dati

**PROGRETTO DI UNA BASE DI DATI
PER LA GESTIONE DI SERVIZI WEB
PER LA PRENOTAZIONE DI
BIGLIETTI A CINEMA**

A.A 2016/2017

Giovanni Mario Russo
Matricola : N46002289

21 dicembre 2016

1 INTRODUZIONE

Una multinazionale di informatica ha recentemente acquisito una serie di società che gestiscono una serie di servizi web per la prenotazione di biglietti a cinema collocati sul territorio svizzero (in particolare ogni società gestisce la prenotazione per i cinema di un singolo cantone).

1.1 SPECIFICHE SUI DATI

Come parte della nuova strategia gestionale, la suddetta società intende fornire ai propri utenti un servizio centralizzato per:

1. La consultazione della programmazione delle proiezioni di film presso i vari cinema;
2. La prenotazione ed acquisto on-line di biglietti.

La società prevede che i servizi sopraelencati siano accessibili dai propri utenti mediante interfacce web. Dal colloquio con il committente e dall'analisi dei processi, sono emerse le seguenti specifiche:

Per gli Utenti devono essere conservate le informazioni anagrafiche, nome e cognome, e quelle di accesso come username, password ed e-mail. La base di dati deve tenere traccia di tutte le prenotazioni con la relativa data ed acquisti di biglietti da parte di utenti per le vari proiezioni con la scelta dei posti. Devono essere memorizzate le sale dei vari cinema che sono contraddistinte da un numero, da una superficie espressa in metri quadrati e una capienza. Ogni cinema ha il suo numero di sale insieme ad informazioni come: nome, indirizzo, città, nazione e recapiti telefonici. Va gestita inoltre, per ogni film, la programmazione (composta da: giorno, orario e prezzo del biglietto) nelle varie

sale dei cinema. Le informazioni che riguardano i film in programmazione sono: titolo, recensione, regia, genere, anno, paese, durata, attori, data di uscita e distributore.

Supposto che nell' arco del prossimo anno la base di dati dovrà gestire la seguente mole di informazioni:

- circa 50 cinema.
- circa 1000 utenti.
- circa 1000 film.

1.2 SPECIFICHE SULLE OPERAZIONI

- Inserimento di un nuovo utente (circa 3 volte al giorno)
- Inserimento di una nuova prenotazione (circa 5 volte al giorno)
- Cancellazione di un Film da parte di un amministratore (circa 1 volta al mese)
- Aggiornamento della programmazione (1 volta al mese)
- Visualizzazione della programmazione da parte degli utenti (circa 20 volte al giorno)

1.3 VINCOLI TECNOLOGICI E POLITICHE DI SICUREZZA

L'architettura prevista per l'applicazione di gestione dei ricoveri è una classica architettura three-tier: a *livello* presentazione sono presenti le interfacce utente dell'applicazione (web o di tipo desktop), a *livello applicazione* sono presenti gli oggetti software che realizzano la vera e propria logica applicativa, infine, a *livello dati* si trovano i DBMS e le informazioni da esso gestite. Il DBMS è installato su un server dotato di 6GB RAM e 4HD che rendono disponibile 1TB di memoria di massa. Esistono poi 3 categorie di utenti che interagiscono con il sistema di base di dati:

- Il *dba* del sistema, che possiede tutti i privilegi possibili sullo schema della base di dati;
- Gli *Utenti* che tramite una applicazione web possono visualizzare le informazioni relative alla programmazione ed effettuare una nuova prenotazione.
- Gli *Amministratori* che tramite un applicazione desktop possono inserire, aggiornare e cancellare informazioni riguardanti film e programmazione, nonché tutte le informazioni relative ai ricoveri e ed alle degenze;

2 PROGETTAZIONE DELLA BASE DI DATI

2.1 ANALISI DELLE SPECIFICHE E RISTRUTTURAZIONE DEI REQUISITI

Al fine di avere una visione più chiara di quelli che sono i requisiti statici richiesti dall'applicazione, si è scelto di effettuare una prima settorializzazione suddividendoli in categorie sulla base di dati.

Informazioni generali

Si vuole progettare una base di dati per una multinazionale di informatica che contenga informazioni relative sia alla prenotazione ed acquisto on-line di biglietti sia alla programmazione delle proiezioni di film.

Informazioni sugli utenti

Per gli Utenti devono essere conservate le informazioni anagrafiche, nome e cognome, e quelle di accesso come username, password ed e-mail.

Informazioni sulle prenotazioni

La base di dati deve tenere traccia di tutte le prenotazioni con la relativa data ed acquisti di biglietti da parte di utenti per le vari proiezioni con la scelta dei posti.

Informazioni sulle sale

Devono essere memorizzate le sale dei vari cinema che sono contraddistinte da un numero, da una superficie espressa in metri quadrati e una capienza. Ogni cinema ha il suo numero di sale insieme ad informazioni come: nome, indirizzo, città, nazione e recapiti telefonici.

Informazioni sulla programmazione

Va gestita inoltre, per ogni film, la programmazione (composta da: giorno, orario e prezzo del biglietto) nelle varie sale dei cinema. Le informazioni che riguardano i film in programmazione sono: titolo, recensione, regia, genere, anno, paese, durata, attori, data di uscita e distributore.

2.2 LA PROGETTAZIONE CONCETTUALE

LO SCHEMA ER PORTANTE

Nella seguente base di dati si sono scelte come entità fondamentali :

PRENOTAZIONE, PROGRAMMAZIONE e CINEMA, rappresentati rispettivamente: l'insieme prenotazioni effettuate presso la biglietteria online, l'insieme degli spettacoli in programma e ovviamente i vari cinema. Le entità PRENOTAZIONE e PROGRAMMAZIONE sono legate dall'associazione SCELTA poichè durante una prenotazione bisogna effettuare una scelta su quale spettacolo andare a vedere, mentre le entità PROGRAMMAZIONE e CINEMA sono legate dall'associazione INFORMAZIONI proprio perchè ogni cinema ha la sua programmazione con informazioni riguardanti gli orari degli spettacoli e il prezzo dei biglietti. Alla luce di queste considerazioni iniziali, un possibile *schema portante* per la nostra base di dati potrebbe essere quello in Figura 1. L'entità PRENOTAZIONE fa riferimento ad un'unica programmazione di un cinema, quindi partecipa all'associazione SCELTA con cardinalità (1,1). L'entità PROGRAMMAZIONE partecipa all'associazione SCELTA con cardinalità (0,N), poichè per una programmazione di un cinema ci possono essere nessuna o piu' prenotazioni. Mentre l'entità PROGRAMMAZIONE partecipa all'asso-

ciazione INFORMAZIONI con cardinalità (1,1), poichè ogni programmazione è associata ad un solo cinema. Sulla base di analoghe considerazioni si ha che l'entità Cinema partecipa all'associazione INFORMAZIONI con cardinalità (1,1).

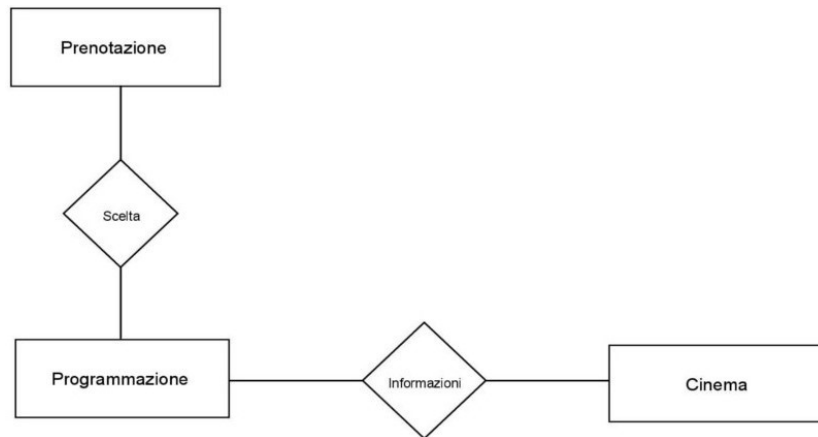


Figura 1: Er Portante

Termine	Descrizione	Termini Collegati
Utente	Un profilo utente è una fonte di dati relativa a tutte le informazioni dell'utente che possono essere impiegate per determinare il comportamento del sistema.	Prenotazione
Prenotazione	Strumento con il quale si possono riservare posti in una sala di un determinato cinema con la scelta di un film con la relativa programmazione.	Utente, Film
Sala	Sezione di un cinema adibita alla proiezione dei Film in programmazione.	Film, Cinema
Cinema	Struttura divisa in sale adibite alla proiezione degli ultimi film in uscita con una determinata programmazione.	Film, Sala, Programmazione
Film	Opera cinematografica e forma di intrattenimento realizzata da artisti tra i quali attori e regista.	Cinema, Sala Prenotazione Programmazione
Regista	responsabile artistico e tecnico di un'opera audiovisiva, dirige gli attori e coordina il set.	Film
Attore	L'attore cinematografico è un attore che usa come mezzo espressivo l'obiettivo di una telecamera. La recitazione è relativa principalmente all'inquadratura di una scena.	Film
Programmazione	Raccolta di informazioni riguardanti la data e gli orari dei vari film in programma in un cinema.	Film, Cinema

Tabella 1 : Glossario Dei Termini

Raffinamento dello schema: lo schema ER finale

Se si vanno ad esaminare le specifiche, ci si accorge che esistono ulteriori entità ed associazioni da prendere in considerazione. Innanzitutto, poichè una prenotazione per esistere deve essere effettuata da un utente è opportuno inse-

rire l'entità **UTENTE** e collegarla a **PRENOTAZIONE** tramite l'associazione **BIGLIETTERIA ONLINE**. Inoltre all'entità programmazione va collegata l'entità film tramite l'associazione **ASSEGNAZIONE**, e siccome al film partecipano degli artisti, vi è un'entità **ARTISTA** collegata a film tramite l'associazione **CASTING**. Da considerare c'è che un'artista può essere un regista od un attore (in rari casi anche entrambi) e quindi esiste una generalizzazione di tipo totale-sovrapposta che lega l'entità padre **ARTISTA** alle l'entità figlie **ATTORE** e **REGISTA**. Infine va aggiunta l'entità **SALA** che è collegata all'entità **PRENOTAZIONE** dall'associazione **VISIONE** e a **CINEMA** dall'associazione **STRUTTURA** e all'entità **PROGRAMMAZIONE** dall'associazione **PROIEZIONE**. Per ciò che riguarda la cardinalità delle associazioni valgono le seguenti considerazioni.

- Un **UTENTE** può solamente essere registrato senza effettuare prenotazioni oppure farne diverse (0,N), mentre una **PRENOTAZIONE** può non avere utenti nel caso di nessuna prenotazione, oppure averne massimo uno per prenotazione (0,1), ovvero il rapporto di cardinalità dell'associazione **BIGLIETTERIA ONLINE** è di tipo uno a molti.
- Una data **PRENOTAZIONE** può fare riferimento ad un'unica programmazione (1,1) mentre una specifica **PROGRAMMAZIONE** può avere nessuna o più prenotazioni (0,N), ovvero il rapporto di cardinalità dell'associazione **SCELTA** è di uno a molti.
- Una data **PRENOTAZIONE** può fare riferimento ad un'unica sala (1,1), mentre una **SALA** può avere nessuna o più prenotazioni (0,N), ovvero il rapporto di cardinalità dell'associazione **POSIZIONE** è di tipo uno a molti.
- Una **PROGRAMMAZIONE** può avere un film (1,1), mentre un **FILM** può avere nessuna programmazione, ad esempio quando un film è appena uscito, oppure diverse (0,N), ovvero il rapporto di cardinalità dell'associazione **ASSEGNAZIONE** è di tipo uno a molti.
- Un **FILM** può avere da uno a più artisti (1,N) mentre un **ARTISTA** può partecipare da uno a più film (1,N), ovvero il rapporto di cardinalità dell'associazione **CASTING** è di tipo molti a molti.
- Una **SALA** può avere un'unica programmazione (1,1), mentre una **PROGRAMMAZIONE** può avere da una a più sale (1,N), ovvero il rapporto di cardinalità dell'associazione **PROIEZIONE** è di tipo uno a molti.
- Un **CINEMA** è strutturato con più sale (1,N), mentre una **SALA** appartiene ad un solo cinema (1,1), ovvero il rapporto di cardinalità dell'associazione **STRUTTURA** è di tipo uno a molti.

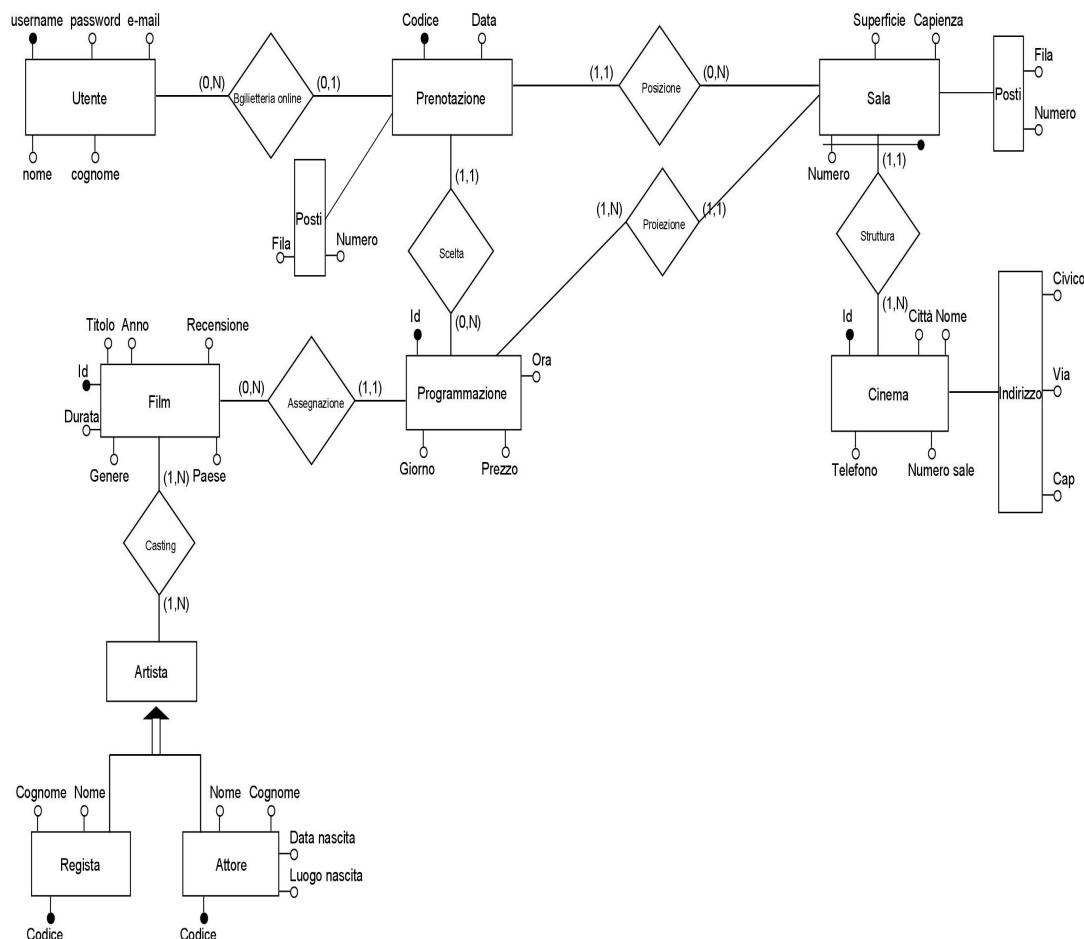


Figura 2: ER concettuale complessivo

2.3 LA PROGETTAZIONE LOGICA

La fase di trasformazione La fase successiva alla progettazione concettuale è quella della progettazione logica. In particolare con riferimento allo schema di Figura 2, la trasformazione dello schema concettuale consiste nell'eliminazione della generalizzazione e nell'eliminazione dell'attributo composto *indirizzo* e di quello multivalore *telefono* di *Cinema*, gli attributi composti scompaiono e gli attributi componenti vengono associati all'entità a cui erano legati, mentre l'attributo multivalore diventa un'entità, avente un solo attributo *Numero*, legata tramite associazione con rapporto di cardinalità uno a molti all'entità *Cinema*. Inoltre gli attributi *posti* di *Prenotazione* e *Sala* sono sia composti che multivalore quindi si è scelto di trasformarli in una entità. Per ciò che concerne la

generalizzazione, questa può essere risolta con la soluzione 2 essendo tale soluzione particolarmente consigliabile quando i vincoli di gerarchia sono di tipo totale-sovrapposta, infatti con questa soluzione si evitano ridondanze dovute a sovrapposizioni. Lo schema er trasformato è riportato in figura 3.

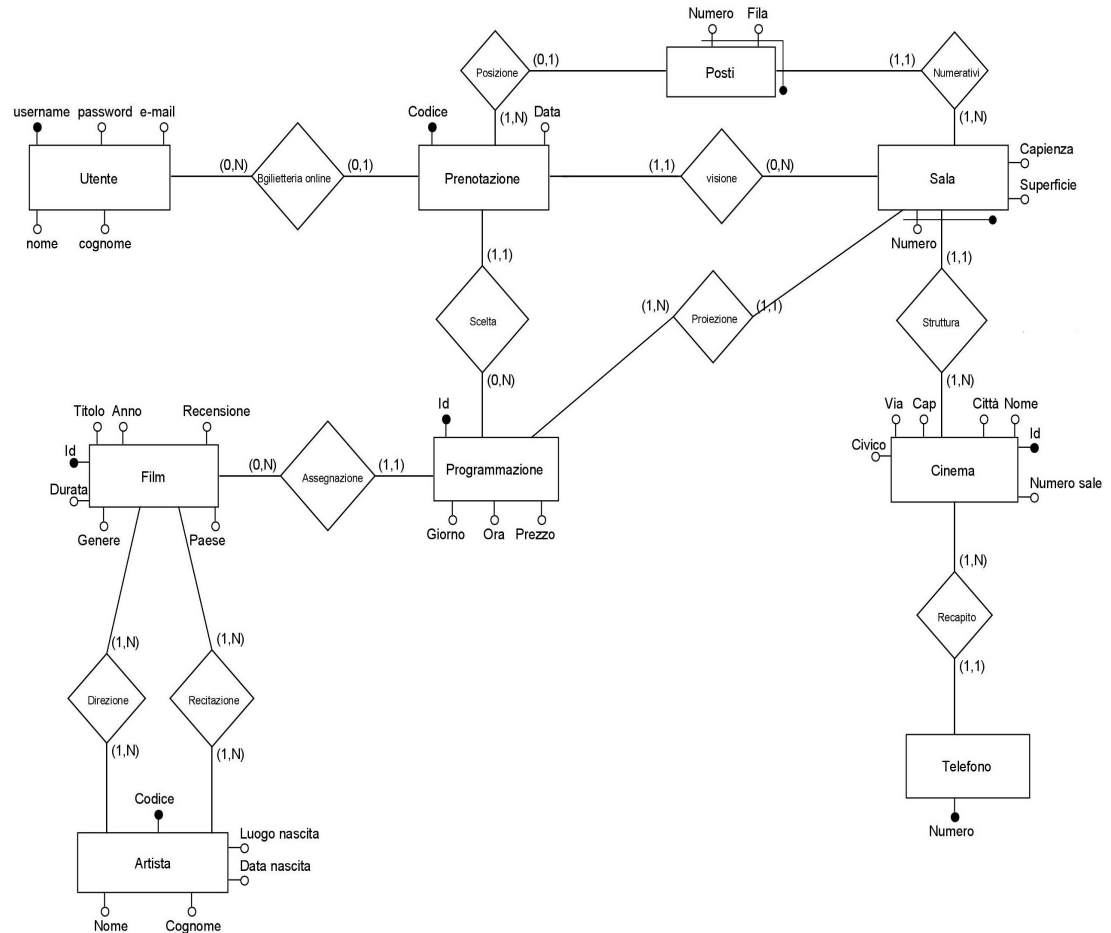


Figura 3: ER Trasformato

La fase di traduzione

1. Ogni entità si trasforma in una relazione avente come attributi gli attributi dell'entità e come chiave primaria l'identificatore dell'entità.
2. Le associazioni BIGLIETTERIA ONLINE, SCELTA, PROIEZIONE, STRUTTURA, RECAPITO, NUMERATIVI, POSIZIONE, VISIONE, ASSEGNAZIONE scompaiano e contemporaneamente, gli identificatori, opportunamente rinominati, dell'entità lato *molti* si aggiungono agli attributi delle relazioni relative all'entità lato *uno* (ovvero username di UTENTE e id di PROGRAMMAZIONE e numero, idCinema di SALA vanno in PRENOTAZIONE, id di PROGRAMMAZIONE e id di CINEMA vanno in

SALA, mentre id di CINEMA va in TELEFONO, e codice di PRENOTAZIONE e numero, idCinema di SALA vanno in POSTI ed id di FILM va in PROGRAMMAZIONE). I nuovi attributi inseriti diventano anche chiavi esterne referenzianti le relazioni relative alle entità lato *molte*.

3. Le associazioni DIREZIONE e RECITAZIONE diventano relazioni aventi come chiave primaria le coppie degli identificatori (id di FILM e codice di ARTISTA) opportunamente rinominati, delle entità che associano. Tali attributi costituiranno anche delle chiavi esterne referenzianti le relazioni relative all'entità da cui provengono.

Schema relazionale completo:

UTENTI (Username, Password, E-mail, Nome, Cognome)

PRENOTAZIONI (Codice, Data, UserUtente:UTENTI, IdProgrammazione:PROGRAMMAZIONI, NumSala:SALE, IdCinema:SALE)

POSTI (NumSala:SALE, IdCinema:SALE, Numero, Fila, CodPrenotazione:PRENOTAZIONI)

PROGRAMMAZIONI (Id, Giorno, Ora, Prezzo, IdFilm:FILM)

FILM (Id, Titolo, Anno, Recensione, Paese, Genere, Durata)

ARTISTI (Codice, Nome, Cognome, DataNascita, LuogoNascita)

CINEMA (Id, Nome, Città, NumeroSale, Via, Civico, Cap)

TELEFONI (Numero, IdCinema:CINEMA)

SALE (Numero, IdCinema:CINEMA, Capienza, Superficie, IdProgrammazione:PROGRAMMAZIONI)

DIREZIONI (IdFilm:FILM, CodArtista:ARTISTI)

RECITAZIONE (IdFilm:FILM, CodArtista:ARTISTI)

Si può infine facilmente verificare che tutte le relazioni dello schema logico ricavato sono in BCNF.

2.4 LA PROGETTAZIONE FISICA

La progettazione fisica di una base di dati è un processo complesso che prevede l'attuazione delle seguenti fasi:

1. dimensionamento fisico della base di dati, sia globale che a livello dei singoli oggetti, in termini di calcolo dello *storage* richiesto su disco;
2. creazione del database;

3. definizione delle politiche di sicurezza e creazione degli utenti / ruoli
4. creazione degli oggetti della base di dati e definizione dei vincoli;
5. creazione di un'istanza;

Di seguito sono descritti i passi seguiti nella fase di progettazione fisica nelle seguenti condizioni operative:

- Installazione della licenza di Oracle 11g;
- Server con 4CPU 3.00GHz, 6GB Ram, 4HD da 500 GB (In RAID5 con 1TB disponibile) con blocchi dati da 5kbyte.
- Sistema operativo del server: Windows 2003 Advanced Server.

DIMENSIONAMENTO FISICO DELLA BASE DI DATI

TABELLA UTENTI:

Attributo	Tipo	Byte	Initial	Next
			1000 occ.	200 occ.
Username	Varchar2 (20)	20	20k	4k
E-Mail	Varchar2 (50)	50	50k	10k
Nome	Varchar2 (50)	50	50k	10k
Cognome	Varchar2 (50)	50	50k	10k
Totale (dati)			170k	24k
Storage	num bloc- chi(i/n)	byte hea- der(i/n)	Initial	Next
Dati + block header	35/5 bloc- chi	140/20	171k	25k

TABELLA PRENOTAZIONI:

Attributo	Tipo	Byte	Initial	Next
			20000 occ.	5000 occ.
Codice	Number (7)	5	100k	25k
Data	Date	7	140k	35k
UserUtente	Varchar2 (20)	20	400k	100k
IdProgrammazione	Number (7)	5	100k	25k
NumSala	Varchar2 (2)	2	40k	10k
IdCinema	Number (7)	5	100k	25k
Totale (dati)			880k	220k
Storage	num bloc- chi(i/n)	byte hea- der(i/n)	Initial	Next
Dati + block header	177/45 blocchi	708/180	881k	221k

TABELLA POSTI:

Attributo	Tipo	Byte	Initial	Next
			50000 occ.	12000 occ.
NumSala	Varchar2(2)	2	100k	24k
IdCinema	Number(7)	5	250k	60k
Numero	Varchar2(2)	2	100k	24k
Fila	Varchar2(1)	1	50k	12k
CodPrenotazione	Number(7)	5	250k	60k
Totale (dati)			750k	180k
Storage	num bloc- chi(i/n)	byte hea- der(i/n)	Initial	Next
Dati + block header	151/37 blocchi	604/148	751k	181k

TABELLA PROGRAMMAZIONI:

Attributo	Tipo	Byte	Initial	Next
			162000occ.	40000 occ.
Id	Number(7)	5	810k	200k
Giorno	Date	7	1134k	280k
Ora	Varchar2(5)	5	810k	200k
Prezzo	Varchar2(6)	6	972k	240k
IdFilm	Number(7)	5	810k	200k
Totale (dati)			4536k	1120k
Storage	num bloc- chi(i/n)	byte hea- der(i/n)	Initial	Next
Dati + block header	908/224 blocchi	3632/896	4536k	1120k

TABELLA FILM:

Attributo	Tipo	Byte	Initial	Next
			1000 occ.	200 occ.
Id	Number(7)	5	5k	1k
Titolo	Varchar2(50)	50	50k	10k
Recensione	Varchar2(256)	256	256k	51.2k
Paese	Varchar2(50)	50	5k	1k
Genere	Varchar2(50)	50	5k	1k
Durata	Varchar2(10)	10	10k	2k
Totale (dati)			331k	67k
Storage	num bloc- chi(i/n)	byte hea- der(i/n)	Initial	Next
Dati + block header	67/14 bloc- chi	268/56	331k	67k

TABELLA ARTISTI:

Attributo	Tipo	Byte	Initial	Next
			30000 occ.	7000 occ.
Codice	Number(7)	5	105k	35k
Nome	Varchar2(50)	50	1500k	350k
Cognome	Varchar2(50)	50	1500k	350k
DataNascita	Date	7	210k	49k
LuogoNascita	Varchar2(50)	50	1500k	350k
Totale (dati)			4860k	1134k
Storage	num bloc- chi(i/n)	byte hea- der(i/n)	Initial	Next
Dati + block header	973/227 blocchi	3892/908	4861k	1135k

TABELLA CINEMA:

Attributo	Tipo	Byte	Initial	Next
			50 occ.	12 occ.
Id	Number(7)	5	0,25k	0,06k
Nome	Varchar2(50)	50	2,5k	0,6k
Città	Varchar2(50)	50	2,5k	0,6k
NumSale	Varchar2(2)	2	0,1k	0,02k
Via	Varchar2(50)	50	2,5k	0,6k
Civico	Varchar2(4)	4	0,2k	0,04k
Cap	Varchar2(5)	5	0,25k	0,06k
Totale (dati)			9k	3k
Storage	num bloc- chi(i/n)	byte hea- der(i/n)	Initial	Next
Dati + block header	2/1 blocchi	8/4	9k	3k

TABELLA TELEFONO:

Attributo	Tipo	Byte	Initial	Next
			100 occ.	20 occ.
Numero	Number(11)	7	0,7k	0,1k
IdCinema	Number(7)	5	0,5k	0,1k
Totale (dati)			2k	1k
Storage	num bloc- chi(i/n)	byte hea- der(i/n)	Initial	Next
Dati + block header	2/1 blocchi	8/4	2k	1k

TABELLA SALE:				
Attributo	Tipo	Byte	Initial	Next
			300 occ.	80 occ.
Numero	Varchar2(2)	2	0,6k	0,1k
IdCinema	Number(7)	5	1,5k	0,4k
Capienza	Varchar2(3)	3	0,9k	0,2k
Superficie	Varchar2(10)	10	3k	0,8k
IdProgrammazione	Number(7)	5	1,5k	0,4k
Totale (dati)			8k	2k
Storage	num bloc- chi(i/n)	byte hea- der(i/n)	Initial	Next
Dati + block header	2/1 blocchi	8/4	8k	2k

TABELLA DIREZIONI:				
Attributo	Tipo	Byte	Initial	Next
			1000 occ.	200 occ.
IdFilm	Number(7)	5	5k	1k
CodArtisti	Number(7)	5	5k	1k
Totale (dati)			10k	2k
Storage	num bloc- chi(i/n)	byte hea- der(i/n)	Initial	Next
Dati + block header	3/1 blocchi	12/4	11k	3k

TABELLA RECITAZIONE:				
Attributo	Tipo	Byte	Initial	Next
			30000 occ.	7000 occ.
IdFilm	Number(7)	5	150k	35k
CodArtisti	Number(7)	5	150k	35k
Totale (dati)			300k	70k
Storage	num bloc- chi(i/n)	byte hea- der(i/n)	Initial	Next
Dati + block header	61/15 bloc- chi	244/60	301k	71k

Si noti che escluse le tabelle FILM, CINEMA e UTENTI le relative occorrenze delle tabelle, sebbene non esplicitamente espresse nelle specifiche fornite, sono state ricavate dai dati disponibili. Per la gestione e l'inserimento di codici/id progressivi sulle tabelle CINEMA, FILM, ARTISTI, PROGRAMMAZIONI e PRENOTAZIONI si farà ricorso ad oggetti di tipo sequenza, istanziabili in oracle con CREATE SEQUENCE. Si tratta di oggetti separati dalle tabelle con i quali introdurre contatori condivisi che permettono di generare sequenze di

numeri interi progressivi.

Tabella Dimensionamento finale:

Oggetto	Nome	Initial	Next
Tabella	UTENTI	171K	25k
Tabella	PRENOTAZIONI	881k	22k
Tabella	POSTI	751k	181k
Tabella	PROGRAM- -MAZIONI	4536k	1120k
Tabella	FILM	331k	67k
Tabella	ARTISTI	4861K	1135K
Tabella	CINEMA	9k	3k
Tabella	TELEFONO	2k	1k
Tabella	SALE	8k	2k
Tabella	DIREZIONI	11k	3k
Tabella	RECITAZIONE	301k	71k
Sequenza	codPre	16b	-
Sequenza	idPro	16b	-
Sequenza	idFlm	16b	-
Sequenza	codArt	16b	-
Sequenza	idCin	16b	-
Database	MULTISALA	16 MB	-

La tabella cui sopra sintetizza i requisiti di memorizzazione della base di dati (la stima per l'occupazione della base di dati che tiene conto dei dati di regime e di quelli previsti di circa 15 MB è stata arrotondata a 16 MB per tenere conto di errori e di altri oggetti come viste, sequenze ed indici non previsti inizialmente).

Configurazione del DB Server

Prima di procedere all'implementazione fisica vera e propria bisogna effettuare un partizionamento della memoria in più blocchi di grandezza differente, ai fini di una corretta gestione dello spazio di memoria disponibile, ogni blocco è adibito al contenimento di un determinato gruppo di file accomunati da analoghe caratteristiche. Si premette che nonostante la disponibilità di 4 Hard Disk da 500Gb (per una dimensione totale di 2TB), la configurazione di questi ultimi in RAID 5 permette lo sfruttamento di 1TB, condizione legata alla relativa politica di gestione di eventuali malfunzionamenti mediante blocchi di parità memorizzati sui singoli dischi. Il partizionamento pensato, considerando che il DBMS utilizzato è ORACLE, sarà del seguente tipo:

- C:[Dimensione 30GB] - partizione dedicata al Sistema Operativo ed ad altro software di base.
- D:[Dimensione 60GB] - partizione dedicata all'installazione del DBMS ORACLE 11g e dei tool di gestione (es. backup/recovery).

- E:[Dimensione 10GB] - partizione dedicata ai Control File di ORACLE
- F:[Dimensione 300GB] - partizione dedicata ai Log File di ORACLE
- G:[Dimensione 600Gb] - partizione dedicata ai Data File di ORACLE

Creazione del database

Si è scelto di creare un database di tipo General Purpose or Transaction processing, e si sono impostate le destinazioni delle strutture di memorizzazione (Control File, Redo Log e Datafile). Si è infine selezionato come DATABASE CHARACTER SET l'Unicode AAL32UTF8 (che supporta i caratteri appartenenti a lingue diverse), ma anche il DATE FORMAT a DD-MON-YY di default per la lingua italiana.

La creazione dei tablespaces

Nel caso in esame sarà creato un tablespace atto a contenere i dati delle varie tabelle. Il codice SQL relativo alla creazione del tablespace è il seguente.

```
CREATE TABLESPACE ts_multisala DATAFILE
'multisala01.dbf' SIZE 16M;
```

Creazione degli utenti e definizione delle politiche di sicurezza

Con riferimento alle specifiche sulle politiche di sicurezza, si è deciso di creare un utente dba, proprietario della base di dati, e 2 ruoli relativi agli utenti e agli amministratori che hanno differenti privilegi sulla base di dati. Lo script per la creazione del dba della base di dati è il seguente:

```
CREATE USER multisala_dba DEFAULT TABLESPACE ts_multisala
IDENTIFIED BY calcio;
GRANT dba, unlimited tablespace TO multisala_dba;
```

Di seguito infine si riportano gli script relativi alla generazione dei 3 ruoli con la definizione dei relativi privilegi.

```
CREATE ROLE utente;
GRANT CONNECT TO utente;
GRANT SELECT ON multisala_dba.PRENOTAZIONI TO utente;
GRANT SELECT ON multisala_dba.PROGRAMMAZIONI TO utente;
GRANT SELECT ON multisala_dba.FILM TO utente;
```

```
CREATE ROLE amministratore;
GRANT CONNECT TO amministratore;
GRANT SELECT, INSERT, DELETE, UPDATE ON
multisala_dba.PRENOTAZIONI
TO amministratore;
GRANT SELECT, INSERT, DELETE, UPDATE ON
```



```

multisala_dba.PROGRAMMAZIONI
TO amministratore;
GRANT SELECT,INSERT,DELETE,UPDATE ON multisala_dba.FILM
TO amministratore;
GRANT SELECT,INSERT,DELETE,UPDATE ON multisala_dba.ARTISTI
TO amministratore;
GRANT SELECT,ALTER ON multisala_dba.idPro TO amministratore;
GRANT SELECT,ALTER ON multisala_dba.idCin TO amministratore;
GRANT SELECT,ALTER ON multisala_dba.idFlm TO amministratore;

```

Creazione degli oggetti della base di dati

Di seguito è riportato lo script per la generazione di tutti gli oggetti della base di dati. Come noto, la creazione dei vincoli di integrità referenziale è ottenuta mediante dei comandi di ALTER TABLE.

```

CREATE TABLE UTENTI(
  Username varchar2(20) ,
  Email varchar2(50) UNIQUE NOT NULL,
  Nome varchar2(50) NOT NULL,
  Cognome varchar2(50) NOT NULL,
  CONSTRAINT CK_UTENTI CHECK(LENGTH(Username)>3),
  CONSTRAINT PK_UTENTI PRIMARY KEY(Username)
)
STORAGE (INITIAL 171K NEXT 25K MINEXTENTS 1 MAXEXTENTS 5
PCTINCREASE 0);

```

```

CREATE TABLE PRENOTAZIONI(
  Codice number(7) ,
  Data date NOT NULL,
  UserUtente varchar2(20) ,
  IdProgrammazione number(7) ,
  NumSala varchar2(2) ,
  IdCinema number(7) ,
  CONSTRAINT PK_PRENOTAZIONI PRIMARY KEY(Codice)
)
STORAGE (INITIAL 881 NEXT 221K MINEXTENTS 1 MAXEXTENTS 5
PCTINCREASE 0);

```

```

CREATE TABLE POSTI(
  NumSala varchar2(2) ,
  IdCinema Number(7) ,
  Numero varchar2(2) ,
  Fila varchar2(1) ,
  CodPrenotazione number(7) ,
  CONSTRAINT PK_POSTI PRIMARY KEY(NumSala , IdCinema , Numero , Fila )
)
STORAGE (INITIAL 751k NEXT 181k MINEXTENTS 1 MAXEXTENTS 5 PCTINCREASE 0);

```

```

CREATE TABLE PROGRAMMAZIONI(
  Id number(7) ,

```

```

    Giorno date NOT NULL,
    Ora varchar2(5) NOT NULL,
    Prezzo varchar2(6) NOT NULL,
    IdFilm number(7) ,
    CONSTRAINT PK_PROGRAMMAZIONI PRIMARY KEY(Id)
)
STORAGE (INITIAL 4536k NEXT 1120k MINEXTENTS 1 MAXEXTENTS 5 PCTINCREASE 0);

CREATE TABLE FILM(
    Id number(7) ,
    Titolo varchar2(50) NOT NULL,
    Recensione varchar2(256) ,
    Paese varchar2(50) ,
    Genere varchar2(50) NOT NULL,
    Durata varchar2(10) NOT NULL,
    CONSTRAINT PK_FILM PRIMARY KEY(Id)
)
STORAGE (INITIAL 331k NEXT 67k MINEXTENTS 1 MAXEXTENTS 5
PCTINCREASE 0);

CREATE TABLE ARTISTI(
    Codice number(7) ,
    Nome varchar2(50) NOT NULL,
    Cognome varchar2(50) NOT NULL,
    DataNascita date NOT NULL,
    LuogoNascita varchar2(50) ,
    CONSTRAINT PK_ARTISTI PRIMARY KEY(Codice)
)
STORAGE (INITIAL 4861k NEXT 1135k MINEXTENTS 1 MAXEXTENTS 5 PCTINCREASE 0);

CREATE TABLE CINEMA(
    Id number(7) ,
    Nome varchar2(50) NOT NULL,
    Città varchar2(50) NOT NULL,
    NumSale varchar2(2) ,
    Via varchar2(50) NOT NULL,
    Civico varchar2(4) NOT NULL,
    Cap varchar2(5) NOT NULL,
    CONSTRAINT PK_CINEMA PRIMARY KEY(Id)
)
STORAGE (INITIAL 9k NEXT 3k MINEXTENTS 1 MAXEXTENTS 5
PCTINCREASE 0);

CREATE TABLE TELEFONI(
    Numero number(11) ,
    IdCinema number(7) ,
    CONSTRAINT PK_TELEFONI PRIMARY KEY(Numero)
)
STORAGE (INITIAL 2K NEXT 1K MINEXTENTS 1 MAXEXTENTS 5
PCTINCREASE 0);

```

```

CREATE TABLE SALE(
Numero varchar2(2),
IdCinema number(7),
Capienza varchar2(3) NOT NULL,
Superficie varchar2(10) NOT NULL,
IdProgrammazione number(7),
CONSTRAINT PK_SALE PRIMARY KEY(Numero, IdCinema)
)
STORAGE (INITIAL 8k NEXT 2k MINEXTENTS 1 MAXEXTENTS 5
PCTINCREASE 0);

CREATE TABLE DIREZIONI(
IdFilm number(7),
CodArtisti number(7),
CONSTRAINT PK_DIREZIONI PRIMARY KEY(IdFilm, CodArtisti)
)
STORAGE (INITIAL 11k NEXT 3k MINEXTENTS 1 MAXEXTENTS 5
PCTINCREASE 0);

CREATE TABLE RECITAZIONE(
IdFilm number(7),
CodArtisti number(7),
CONSTRAINT PK_DIREZIONI PRIMARY KEY(IdFilm, CodArtisti)
)
STORAGE (INITIAL 301k NEXT 71k MINEXTENTS 1 MAXEXTENTS 5
PCTINCREASE 0);

CREATE SEQUENCE codPre
START WITH 1
INCREMENT BY 1
NOMAXVALUE
NOCYCLE
;

CREATE SEQUENCE idPro
START WITH 1
INCREMENT BY 1
NOMAXVALUE
NOCYCLE
;

CREATE SEQUENCE idFlm
START WITH 1
INCREMENT BY 1
NOMAXVALUE
NOCYCLE
;

CREATE SEQUENCE codArt

```

```
START WITH 1
INCREMENT BY 1
NOMAXVALUE
NOCYCLE
;
```

```
CREATE SEQUENCE idCin
START WITH 1
INCREMENT BY 1
NOMAXVALUE
NOCYCLE
;
```

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-Mon-YYYY';
```

```
ALTER TABLE PRENOTAZIONI
ADD CONSTRAINT FK_USER_PRE FOREIGN KEY (UserUtente)
REFERENCES UTENTI(Username)
ON DELETE CASCADE;
```

```
ALTER TABLE PRENOTAZIONI
ADD CONSTRAINT FK_PRO_PRE FOREIGN KEY (IdProgrammazione)
REFERENCES PROGRAMMAZIONI(Id)
ON DELETE CASCADE;
```

```
ALTER TABLE PRENOTAZIONI
ADD CONSTRAINT FK_SAL_PRE FOREIGN KEY (NumSala, IdCinema)
REFERENCES SALE(Numero, IdCinema)
ON DELETE CASCADE;
```

```
ALTER TABLE POSTI
ADD CONSTRAINT FK_SAL_POS FOREIGN KEY (NumSala, IdCinema)
REFERENCES SALE(Numero, IdCinema)
ON DELETE SET NULL;
```

```
ALTER TABLE POSTI
ADD CONSTRAINT FK_PRE_POS FOREIGN KEY (CodPrenotazione)
REFERENCES PRENOTAZIONI(Codice)
ON DELETE SET NULL;
```

```
ALTER TABLE PROGRAMMAZIONI
ADD CONSTRAINT FK_FLM_PRO FOREIGN KEY (IdFilm)
REFERENCES FILM(Id)
ON DELETE CASCADE;
```

```
ALTER TABLE TELEFONI
ADD CONSTRAINT FK_CIN_TEL FOREIGN KEY (IdCinema)
REFERENCES CINEMA(Id)
ON DELETE CASCADE;
```

```

ALTER TABLE SALE
ADD CONSTRAINT FK_CIN_SAL FOREIGN KEY (IdCinema)
REFERENCES CINEMA(Id)
ON DELETE CASCADE;

```

```

ALTER TABLE SALE
ADD CONSTRAINT FK_PRO_SAL FOREIGN KEY (IdProgrammazione)
REFERENCES PROGRAMMAZIONI(Id)
ON DELETE SET NULL;

```

```

ALTER TABLE DIREZIONI
ADD CONSTRAINT FK_FLM_DIR FOREIGN KEY (IdFilm)
REFERENCES FILM(Id)
ON DELETE CASCADE;

```

```

ALTER TABLE DIREZIONI
ADD CONSTRAINT FK_ART_DIR FOREIGN KEY (CodArtisti)
REFERENCES ARTISTI(Codice)
ON DELETE CASCADE;

```

```

ALTER TABLE RECITAZIONE
ADD CONSTRAINT FK_FLM_REC FOREIGN KEY (IdFilm)
REFERENCES FILM(Id)
ON DELETE CASCADE;

```

```

ALTER TABLE RECITAZIONE
ADD CONSTRAINT FK_ART_REC FOREIGN KEY (CodArtisti)
REFERENCES ARTISTI(Codice)
ON DELETE CASCADE;

```

```

SELECT Titolo
FROM PROGRAMMAZIONI p join FILM f on p.IdFilm=f.Id
WHERE GIORNO='13-OTT-2017' AND ORA = '21:30'

```

```

SELECT*
FROM PRENOTAZIONI
WHERE UserUtente='PAKY'

```

Il popolamento della base di dati

Il popolamento della base di dati avviene attraverso gli statement SQL di insert oppure attraverso opportune procedure di *import* che automatizzano la creazione dell'istanza.

3 CENNI ALLA PROGETTAZIONE DELLE APPLICAZIONI

3.1 IL LIVELLO DATI

```

create or replace trigger prenotazione_bef_ins

```

```

before insert on Prenotazioni
for each row
DECLARE bool number (2);
errore_ins exception;
BEGIN
select count (*) into bool
from POSTI
where NumSala=:new.NumSala and CodPrenotazione IS NULL;
if bool<1 then raise errore_ins;
end if;
exception
when errore_ins then raise_application_error(-10001,'Posti_esauriti');
END;

```