

# Basi di Dati

## Oracle SQLPLUS - Esercitazione n. 3

Questa esercitazione è composta da due parti. La parte I contiene alcune interrogazioni da risolvere usando il linguaggio SQL su una base di dati già esistente. La parte II ha come obiettivo quello di creare, dato lo schema logico di una base di dati, appositi script di creazione e popolamento della base di dati e di scrivere ed eseguire alcuni comandi di aggiornamento e cancellazione utilizzando il linguaggio SQL.

### PARTE I

#### Passi preliminari per lo svolgimento delle esercitazioni

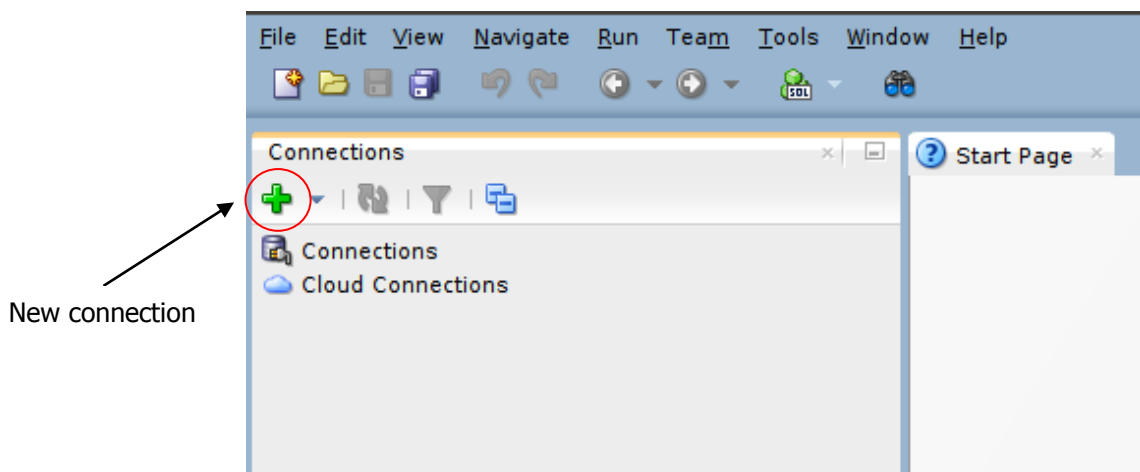
La finalità di queste esercitazioni è quella di scrivere alcune interrogazioni in SQL e di eseguirle su un database Oracle.

#### Connessione al server oracle

Le interrogazioni SQL sono eseguite tramite il software SQL Developer che permette di connettersi alla base di dati Oracle e sottomettere delle interrogazioni SQL.

##### 1) Connessione alla base di dati

- Aprire il programma Oracle SQL Developer
- Cliccare su New connection

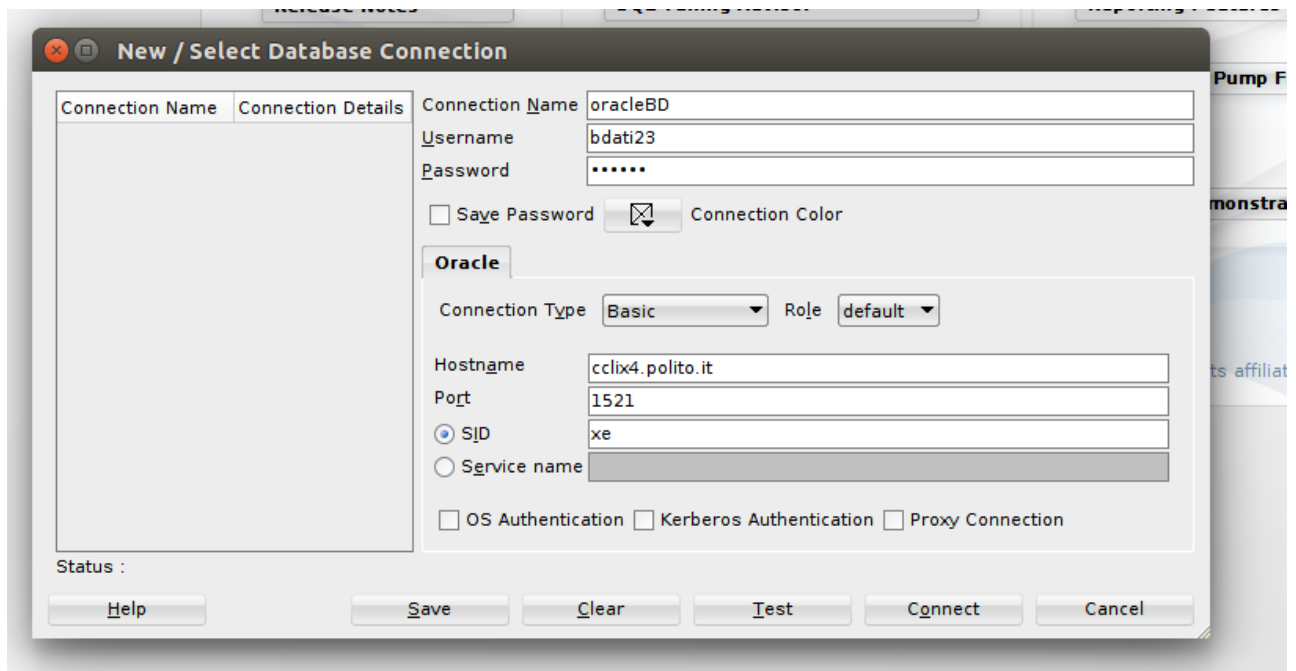


##### 2) Login

Autenticarsi inserendo i seguenti dati:

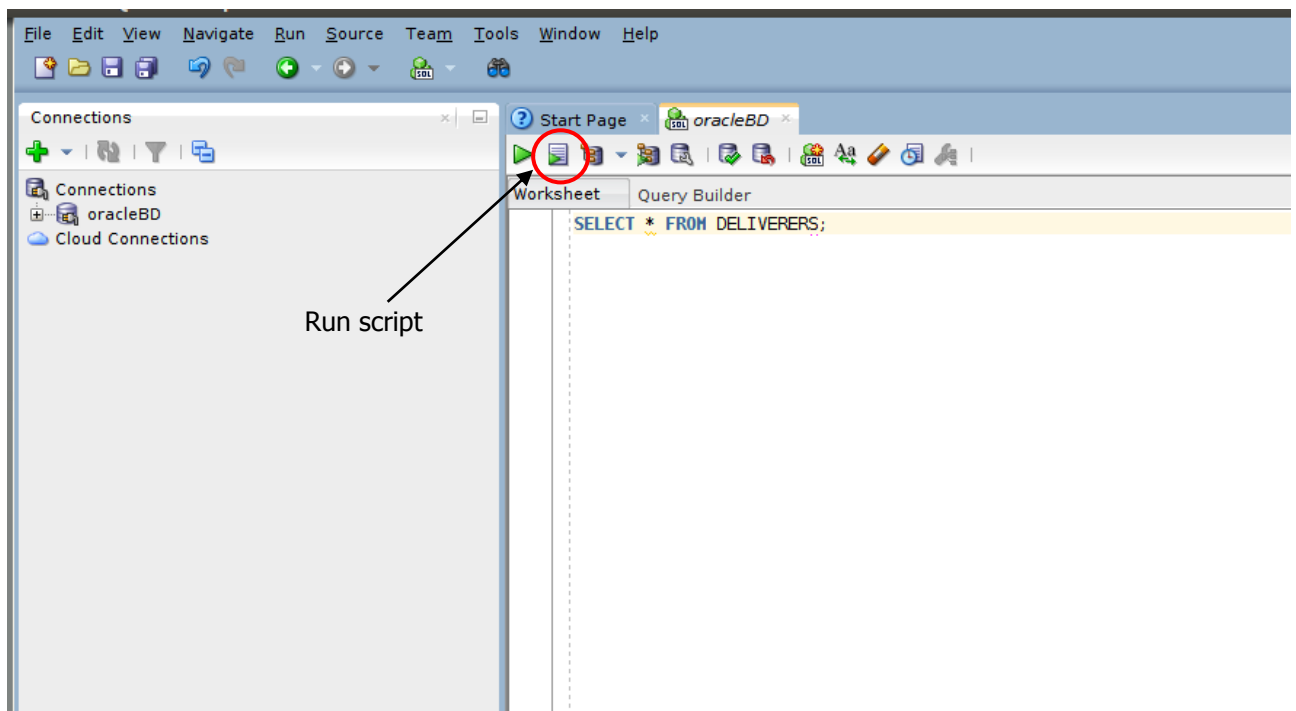
- Connection name: oracleBD
- Username: bdatiXY
  - XY indica le ultime due cifre del numero del pc utilizzato
- Password: oracXY
  - XY indica le ultime due cifre del numero del pc utilizzato
- Hostname: cclix4.polito.it
- Port: 1521
- SID: xe

Ad esempio, collegandosi dalla macchina numero 23 del laboratorio, usare come username bdati23 e come password orac23.



### **Scrittura ed esecuzione delle interrogazioni SQL**

Scrivere l'interrogazione SQL da eseguire nell'area di lavoro (Worksheet) e eseguire l'interrogazione/script premendo il tasto "Run script".



# 1. Descrizione del Database *Delivery*

Il database *Delivery* raccoglie informazioni relative alle attività svolte da una ditta di fattorini che svolge consegne e ritiri di merci per diverse aziende.

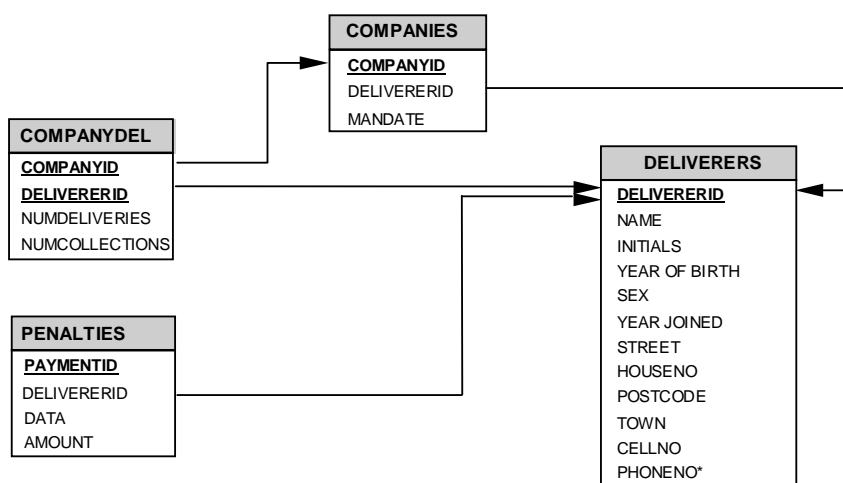
La tabella DELIVERERS contiene i dati anagrafici dei fattorini. In particolare, per ogni fattorino sono disponibili il codice identificativo (DELIVERERID), il nome, le iniziali, l'anno di nascita, il sesso, l'anno di inizio attività, la via, il numero civico, la città, il codice postale del luogo di residenza, il numero di cellulare e l'interno dell'ufficio in cui è dislocato.

Nella tabella COMPANYDEL sono riportati i dati relativi alle consegne e ai ritiri fatti da ogni fattorino presso le varie aziende (identificate dal codice COMPANYID). Per ogni coppia fattorino-azienda per cui è avvenuta almeno una consegna o un ritiro sono noti il numero di consegne (NUMDELIVERIES) e di ritiri (NUMCOLLECTIONS) effettuati.

La tabella PENALTIES raccoglie le multe ricevute dai fattorini. Per ogni multa vengono memorizzati il codice della multa (PENALTYID), il codice del fattorino, la data e l'importo da pagare.

Nella tabella COMPANIES per ogni azienda è noto il codice identificativo dell'azienda (COMPANYID) e il codice identificativo del referente ad essa assegnato (un fattorino) e il numero di mandati per cui il referente attuale ha ricoperto tale carica (MANDATE).

La struttura della base di dati è riportata nella figura seguente e i dati contenuti nelle tabelle sono riportati nella Sezione 2.



## 2. Contenuto delle Tabelle del Database *Delivery*

3. La chiave primaria è sottolineata. I campi che possono assumere il valore nullo sono contrassegnati dal simbolo \*.

Tabella **DELIVERERS**

<u>DELIVERERID</u>	NAME	INITIALS	YEAR_OF_BIRTH	SEX	YEAR JOINED	STREET	HOUSENO	POSTCODE	TOWN	CELLNO	PHONENO*
2	Everett	R	1948	M	1975	Stoney Road	43	3575NH	Stratford	070-237893	2411
6	Parmenter	R	1964	M	1977	Haseltine Lane	80	1234KK	Stratford	070-476537	8467
7	Wise	GWS	1963	M	1981	Edgecombe Way	39	9758VB	Stratford	070-347689	NULL
8	Newcastle	B	1962	F	1980	Station Road	4	6584WO	Inglewood	070-476573	2983
27	Collins	DD	1964	F	1983	Long Drive	804	8457DK	Eltham	079-234857	2513
28	Collins	C	1963	F	1983	Old main Road	10	1294QK	Midhurst	010-659599	NULL
39	Bishop	D	1956	M	1980	Eaton Square	78	9629CD	Stratford	070-393435	NULL
44	Baker	E	1963	M	1980	Lewis Street	23	4444LJ	Inglewood	070-368753	1124
57	Brown	M	1971	M	1985	Edgecombe Way	16	4377CB	Stratford	070-473458	6409
83	Hope	PK	1956	M	1982	Magdalene Road	16a	1812UP	Stratford	070-353548	1608
95	Miller	P	1934	M	1972	High Street	33a	5746OP	Douglas	070-867564	NULL
100	Parmenter	P	1963	M	1979	Haseltine Lane	80	1234KK	Stratford	070-476537	6524
104	Moorman	D	1970	F	1984	Stout Street	65	9437AO	Eltham	079-987571	7060
112	Bailey	IP	1963	F	1984	Vixen Road	8	6392LK	Plymouth	010-54874	1319

Tabella **COMPANYDEL**

<u>COMPANYID</u>	<u>DELIVERERID</u>	NUMDELIVERIES	NUMCOLLECTIONS
1	2	4	8
1	6	9	1
1	8	0	1
1	44	7	5
1	57	5	0
1	83	3	3
2	8	4	4
2	27	11	2
2	104	8	4
2	112	4	8

Tabella **PENALTIES**

<u>PAYMENTID</u>	<u>DELIVERERID</u>	DATA	AMOUNT
1	6	12/08/1980	100
2	44	05/05/1981	75
3	27	10/09/1983	100
4	104	08/12/1984	50
5	44	08/12/1980	25
6	8	08/12/1980	25
7	44	30/12/1982	30
8	27	12/11/1984	75

Tabella **COMPANIES**

<u>COMPANYID</u>	<u>DELIVERERID</u>	MANDATE
1	6	first
2	27	second

### 3. Query

1. Trovare per ogni fattorino che ha preso almeno due multe il codice identificativo del fattorino, la data della prima multa e la data dell'ultima multa che ha preso.
2. Trovare il codice identificato del fattorino che ha ricevuto il numero massimo di multe.
3. Trovare per ogni fattorino che ha preso almeno una multa il codice identificativo del fattorino, la data in cui ha preso l'ultima multa e l'ammontare di tale multa.
4. Trovare l'identificativo delle aziende presso cui si sono recati più del 30% dei fattorini presenti nella base di dati (nota: i fattorini "recatisi" presso un'azienda sono quelli che hanno fatto almeno una consegna o un ritiro presso l'azienda in esame).

## PARTE II

### Passi preliminari per lo svolgimento dell'esercitazione

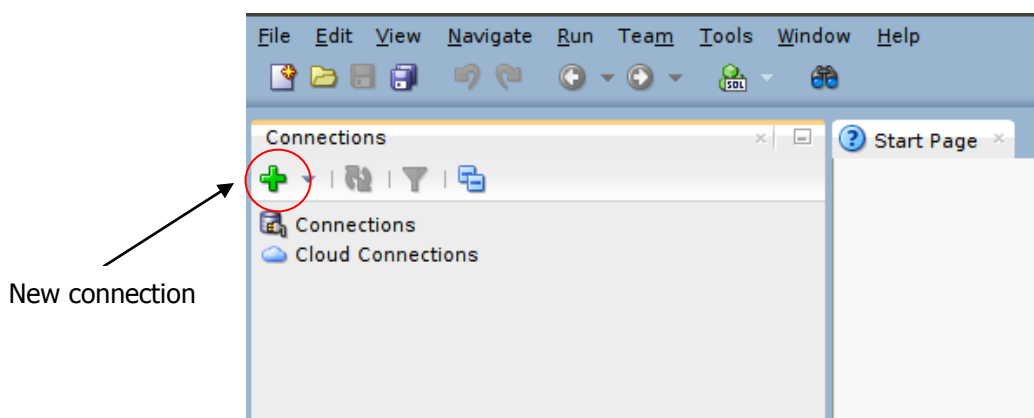
La finalità di questa seconda parte dell'esercitazione è quella di creare, dato lo schema logico di una base di dati, appositi script di creazione e popolamento della base di dati, e di scrivere ed eseguire alcuni comandi di aggiornamento e cancellazione utilizzando il linguaggio SQL.

#### Connessione al server oracle

L'esecuzione degli script e dei comandi SQL sono eseguite tramite il software SQL Developer che permette di connettersi alla base di dati Oracle e eseguire script contenenti comandi SQL o singoli comandi SQL.

#### 1) **Connessione alla base di dati**

- Aprire il programma Oracle SQL Developer
- Cliccare su New connection

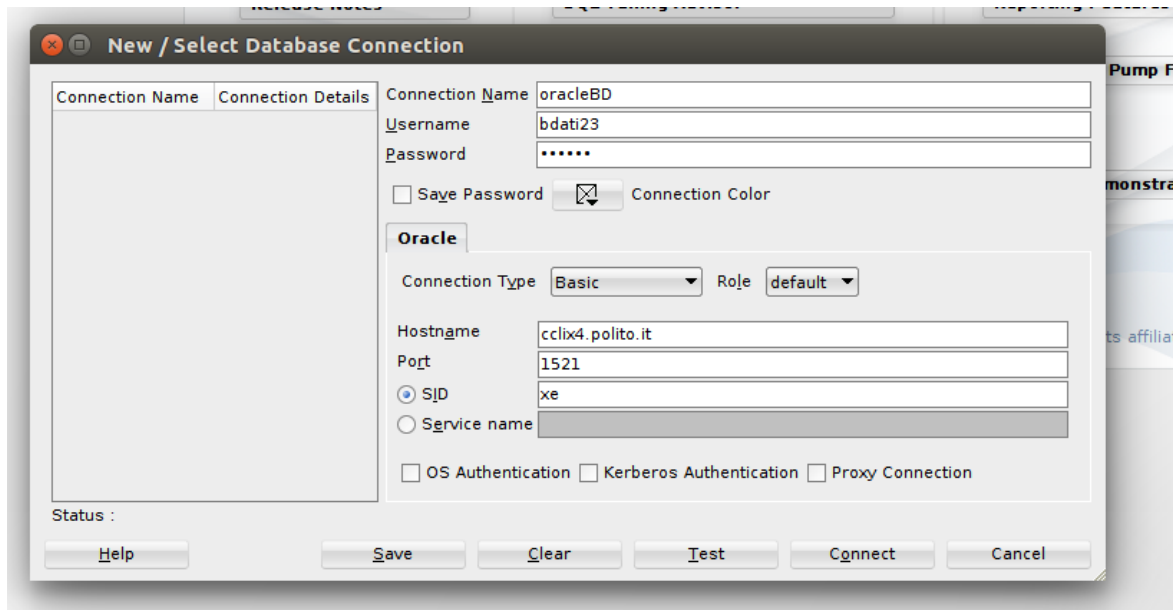


#### 2) **Login**

Autenticarsi inserendo i seguenti dati:

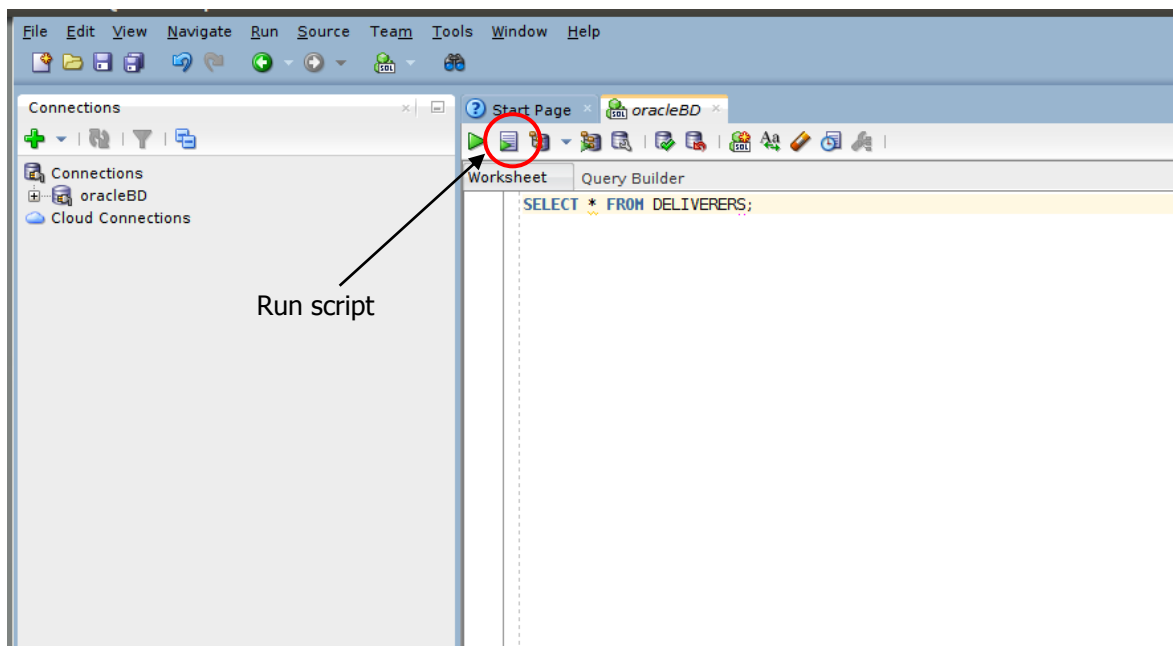
- Connection name: oracleBD
- Username: bdatiXY
  - XY indica le ultime due cifre del numero del pc utilizzato
- Password: oracXY
  - XY indica le ultime due cifre del numero del pc utilizzato
- Hostname: cclix4.polito.it
- Port: 1521
- SID: xe

Ad esempio, collegandosi dalla macchina numero 23 del laboratorio, usare come username bdati23 e come password orac23.



### **Scrittura ed esecuzione di script SQL**

Scrivere lo script SQL da eseguire nell'area di lavoro (Worksheet) e eseguirlo premendo il tasto "Run script". Usando i comandi presenti sotto la voce File è possibile salvare uno script o caricarne uno già esistente.



## 1. Descrizione della base di dati *Palestra*

La base di dati da realizzare è relativa alle attività di una palestra ed è caratterizzata dal seguente schema logico (le chiavi primarie sono sottolineate e le chiavi esterne sono in corsivo):

ISTRUTTORE (CodFisc, Nome, Cognome, DataNascita, Email, Telefono\*)

CORSI (CodC, Nome, Tipo, Livello)

PROGRAMMA (CodFisc, Giorno, OraInizio, Durata, *CodC*, Sala)

Per ogni istruttore è noto il codice fiscale, il nome, il cognome, la data di nascita, l'indirizzo e-mail e il numero di telefono. Per ogni corso è noto il codice, il nome, il tipo (es. attività musicale) e il livello (un numero compreso tra 1 e 4). Il programma dei corsi riporta il giorno della settimana (ad esempio lunedì, martedì, ecc.) e l'ora di inizio in cui ogni istruttore svolge una lezione di un determinato corso e la durata in minuti della lezione. Per ogni lezione programmata è noto il numero della sala in cui si svolge.

Tabella ISTRUTTORE

<u>CodFisc</u>	Nome	Cognome	DataNascita	Email	Telefono
<b>SMTPLA80N31B791Z</b>	Paul	Smith	31/12/1980	<a href="mailto:p.smith@email.it">p.smith@email.it</a>	NULL
<b>KHNJHN81E30C455Y</b>	John	Johnson	30/5/1981	<a href="mailto:j.johnson@email.it">j.johnson@email.it</a>	+2300110303444
<b>AAAGGG83E30C445A</b>	Peter	Johnson	30/5/1981	<a href="mailto:p.johnson@email.it">p.johnson@email.it</a>	+2300110303444

Tabella CORSI

<u>CodC</u>	Nome	Tipo	Livello
<b>CT100</b>	Spinning principianti	Spinning	1
<b>CT101</b>	Ginnastica e musica	Attività musicale	2
<b>CT104</b>	Spinning professionisti	Spinning	4

Tabella PROGRAMMA

<u>CodFisc</u>	<u>Giorno</u>	<u>OrarioInizio</u>	Durata	<i>CodC</i>	Sala
<b>SMTPLA80N31B791Z</b>	Lunedì	10:00	45	CT100	S1
<b>SMTPLA80N31B791Z</b>	Martedì	11:00	45	CT100	S1
<b>SMTPLA80N31B791Z</b>	Martedì	15:00	45	CT100	S2
<b>KHNJHN81E30C455Y</b>	Lunedì	10:00	30	CT101	S2
<b>KHNJHN81E30C455Y</b>	Lunedì	11:30	30	CT104	S2
<b>KHNJHN81E30C455Y</b>	Mercoledì	9:00	60	CT104	S1

Figura 1. Contenuto che deve avere la base di dati dopo l'esecuzione dello script *popolaDB.sql* creato durante l'esercitazione

## 2. Script

1. Creare uno script SQL (*creaDB.sql*) con le istruzioni (CREATE TABLE) per la creazione della base di dati corrispondente allo schema logico riportato nella sezione 1 del testo.

In particolare:

- definire tutte e tre le tabelle, scegliendo il tipo ritenuto più opportuno per gli attributi presenti nelle tabelle. Porre particolare attenzione alla definizione della chiave primaria e alla definizione dei vincoli di integrità referenziale
- specificare eventuali vincoli di dominio e/o di tupla ritenuti appropriati. **Nota.** In Oracle i vincoli di dominio si definiscono alla fine della create table usando la sintassi:

CONSTRAINT nome\_vincolo CHECK (vincolo)

Esempio. Si consideri un'ipotetica tabella che contiene il campo Voto. Per imporre che il dominio di Voto sia tra 0 e 30 si deve scrivere al fondo della create table della tabella che contiene l'attributo Voto il comando

CONSTRAINT vincolo\_voti CHECK (Voto>=0 and Voto<=30)

- scegliere opportunamente le politiche di gestione dei vincoli selezionando quelle più idonee al contesto. **Nota. Oracle non gestisce l'opzione ON UPDATE CASCADE (la politica di default applicata da Oracle per l'update è NO ACTION).** Quindi non specificare nulla per la parte di gestione dei vincoli per la parte relativi agli UPDATE. Gestire invece la parte relativa alle cancellazioni (DELETE) decidendo se usare la politica ON DELETE CASCADE o ON DELETE NO ACTION.

**Nota.** Fare attenzione all'ordine con il quale vengono create le tabelle. Creare prima le tabelle referenziate e solo in seguito quelle referenzianti.

2. Creare uno script SQL (*popolaDB.sql*) contenente le istruzioni di inserimento (INSERT) necessarie per il popolamento della base di dati creata al punto precedente. Lo script deve contenere le istruzioni di inserimento necessarie per ottenere un'istanza della base di dati contenente gli stessi dati riportati nelle tabelle presenti in figura 1.

**Nota.** L'ordine di esecuzione delle INSERT è importante. Usare l'ordine corretto al fine di evitare violazioni dei vincoli di integrità referenziale.

3. Testare gli script di creazione e popolamento scritti ai punti precedenti.

**Nota.** Le tre tabelle potrebbero già esistere nella base di dati nel caso in cui qualche altro studente abbia creato le tabelle durante l'orario di laboratorio precedente al vostro. In tal caso, prima di eseguire i vostri script, eseguite i seguenti comandi per cancellare le tabelle create da altri:

- DROP TABLE PROGRAMMA;
- DROP TABLE CORSI;
- DROP TABLE ISTRUTTORE;



4. Scrivere in SQL e eseguire i seguenti comandi di aggiornamento, uno per volta, e verificare cosa succede nella base di dati.
  - 4.1. Aggiornare il numero di telefono dell'istruttore identificato dal codice fiscale 'KHNJHN81E30C455Y' impostandolo al valore '+390112333551'.
  - 4.2. Aggiornare la base di dati in modo tale da spostare nella sala 'S4' tutte le lezioni in programma presso la sala 'S2'.
  - 4.3. Eliminare dalla tabella CORSI tutti i corsi che sono in programma una sola volta a settimana (ossia che sono presenti una sola volta nella tabella PROGRAMMA). Che effetto ha l'esecuzione di questo comando sulla tabella CORSI? Che effetto ha sul contenuto della tabella PROGRAMMA? L'effetto dell'operazione sulle due tabelle è legato alla politica di gestione delle violazioni scelta durante la creazione delle tabelle?
  - 4.4. Inserire un nuovo corso, con le seguenti caratteristiche, nella tabella CORSI: CodC='CT120', Nome='Spinning estremo', Tipo='Spinning', Livello=10. L'esecuzione del comando genera degli errori? Si può inserire un corso di livello 10? **Nota.** Guardare le specifiche relative al dominio ammissibile per l'attributo livello riportato nella sezione 1 del testo.
  - 4.5. Eliminare l'istruttore con codice fiscale pari a 'SMTPLA80N31B791Z'. Che effetto ha questa operazione sulle tabelle ISTRUTTORE e PROGRAMMA? L'esito dell'esecuzione dell'istruzione come è legata alla politica di gestione delle violazioni imposta durante la creazione delle tabelle?

## 4. Soluzioni

### PARTE I

1. Trovare per ogni fattorino che ha preso almeno due multe il codice identificativo del fattorino, la data della prima multa e la data dell'ultima multa che ha preso.

```
SELECT DELIVEREDID, MIN(DATA), MAX(DATA)
FROM PENALTIES
GROUP BY DELIVEREDID
HAVING COUNT(*)>=2;
```

2. Trovare il codice identificato del fattorino che ha ricevuto il numero massimo di multe.

```
SELECT DELIVERERID
FROM PENALTIES
GROUP BY DELIVERERID
HAVING COUNT(*) = (
    SELECT MAX(NumPenalties)
    FROM (SELECT DELIVERERID, COUNT(*) AS NumPenalties
          FROM PENALTIES
          GROUP BY DELIVERERID) TOTMULTEDELIVERERS
);
```

Oppure

```
SELECT DELIVERERID
FROM (SELECT DELIVERERID, COUNT(*) AS NumPenalties
      FROM PENALTIES
      GROUP BY DELIVERERID) TOTMULTEDELIVERERS1
WHERE NumPenalties = (
    SELECT MAX(NumPenalties)
    FROM (SELECT DELIVERERID, COUNT(*) AS NumPenalties
          FROM PENALTIES
          GROUP BY DELIVERERID) TOTMULTEDELIVERERS2
);
```

3. Trovare per ogni fattorino che ha preso almeno una multa il codice identificativo del fattorino, la data in cui ha preso l'ultima multa e l'ammontare di tale multa.

```
SELECT P1.DELIVEREDID, DATA, AMOUNT
FROM PENALTIES P1
WHERE P1.DATA = (SELECT MAX(DATA)
                 FROM PENALTIES P2
                 WHERE P2.DELIVEREDID=P1.DELIVEREDID);
```

4. Trovare l'identificativo delle aziende presso cui si sono recati più del 30% dei fattorini presenti nella base di dati (nota: i fattorini "recatisi" presso un'azienda sono quelli che hanno fatto almeno una consegna o un ritiro presso l'azienda in esame).

```
SELECT COMPANYID
FROM COMPANYDEL
GROUP BY COMPANYID
HAVING COUNT(*) > ( SELECT 0.30*COUNT(*)
                     FROM DELIVERERS );
```

## PARTE II

1. Creare uno script SQL *creaDB.sql* con le istruzioni per la creazione della base di dati corrispondente allo schema logico riportato nella Sezione 1.

In particolare:

- specificare eventuali vincoli di dominio e/o di tupla appropriati

I vincoli di integrità referenziale presi in considerazione riguardano la presenza nella tabella Programma degli attributi CodC, identificatore univoco della tabella Corsi, e CodFisc, identificatore univoco della tabella Istruttore.

```
FOREIGN KEY (CodFisc)
  REFERENCES Istruttore(CodFisc)
FOREIGN KEY (CodC)
  REFERENCES Corsi(CodC)
```

I vincoli di dominio nel nostro caso riguardano l'attributo Livello (tabella Corsi):

```
CONSTRAINT chk_Livello CHECK (Livello>=1 and Livello<=3)
```

- scegliere opportunamente le politiche di gestione dei vincoli più idonee al contesto

Nel nostro caso come politica di gestione dei vincoli è stato scelto di usare le opzioni ON DELETE CASCADE che propaga l'eventuale operazione di cancellazione sulla tabella referenziata anche alla tabella referenziante.

Se esistono già, perché qualcuno le ha già create, elimino le tre tabelle

```
-- drop tables list
DROP TABLE Programma;
DROP TABLE Corsi;
DROP TABLE Istruttore;
```

-- creazione tabelle

```
CREATE TABLE Istruttore (  
  CodFisc CHAR(20) ,  
  Nome CHAR(50) NOT NULL ,  
  Cognome CHAR(50) NOT NULL ,  
  DataNascita DATE NOT NULL ,  
  Email CHAR(50) NOT NULL ,  
  Telefono CHAR(20) NULL ,  
  PRIMARY KEY (CodFisc)  
);
```

```
CREATE TABLE Corsi (  
  CodC CHAR(10) ,  
  Nome CHAR(50) NOT NULL ,  
  Tipo CHAR(50) NOT NULL ,  
  Livello SMALLINT NOT NULL,  
  PRIMARY KEY (CodC),  
  CONSTRAINT chk_Livello CHECK (Livello>=1 and Livello<=4)  
);
```

```
CREATE TABLE Programma (  
  CodFisc CHAR(20) NOT NULL ,  
  Giorno CHAR(15) NOT NULL ,  
  OraInizio CHAR(5) NOT NULL ,  
  Durata SMALLINT NOT NULL ,  
  Sala CHAR(5) NOT NULL,  
  CodC CHAR(10) NOT NULL,  
  PRIMARY KEY (CodFisc,Giorno,OraInizio),  
  FOREIGN KEY (CodFisc)  
    REFERENCES Istruttore(CodFisc) ON DELETE CASCADE,  
  FOREIGN KEY (CodC)  
    REFERENCES Corsi(CodC) ON DELETE CASCADE  
);
```

2. Creare uno script SQL popolaDB.sql con le istruzioni per il popolamento della base di dati creata al punto precedente.

-- insert data records

```
INSERT INTO Istruttore (CodFisc,Nome,Cognome,DataNascita,Email, Telefono)  
VALUES ('SMTPLA80N31B791Z','Paul','Smith',TO_DATE('31/12/1980',  
'DD/MM/YYYY'),'p.smith@email.it',NULL);
```

```
INSERT INTO Istruttore (CodFisc,Nome,Cognome,DataNascita,Email, Telefono)  
VALUES ('KHNJHN81E30C455Y','John','Johnson',TO_DATE('30/5/1981',  
'DD/MM/YYYY'),'j.johnson@email.it','+2300110303444');
```

```
INSERT INTO Istruttore (CodFisc,Nome,Cognome,DataNascita,Email, Telefono)
```

```
VALUES ('AAAGGG83E30C445A','Peter','Johnson',TO_DATE('30/5/1981',
'DD/MM/YYYY'),'p.johnson@email.it','+2300110303444');
```

```
INSERT INTO Corsi (CodC,Nome,Tipo,Livello)
VALUES ('CT100','Spinning principianti','Spinning ',1);
```

```
INSERT INTO Corsi (CodC,Nome,Tipo,Livello)
VALUES ('CT101','Ginnastica e musica','Attività musicale',2);
```

```
INSERT INTO Corsi (CodC,Nome,Tipo,Livello)
VALUES ('CT104','Spinning professionisti','Spinning',4);
```

```
INSERT INTO Programma (CodFisc,Giorno,OraInizio,Durata,Sala,CodC)
VALUES ('SMTPLA80N31B791Z','Lunedì','10:00',45,'S1','CT100');
```

```
INSERT INTO Programma (CodFisc,Giorno,OraInizio,Durata,Sala,CodC)
VALUES ('SMTPLA80N31B791Z','Martedì','11:00',45,'S1','CT100');
```

```
INSERT INTO Programma (CodFisc,Giorno,OraInizio,Durata,Sala,CodC)
VALUES ('SMTPLA80N31B791Z','Martedì','15:00',45,'S2','CT100');
```

```
INSERT INTO Programma (CodFisc,Giorno,OraInizio,Durata,Sala,CodC)
VALUES ('KHNJHN81E30C455Y','Lunedì','10:00',30,'S2','CT101');
```

```
INSERT INTO Programma (CodFisc,Giorno,OraInizio,Durata,Sala,CodC)
VALUES ('KHNJHN81E30C455Y','Lunedì','11:30',30,'S2','CT104');
```

```
INSERT INTO Programma (CodFisc,Giorno,OraInizio,Durata,Sala,CodC)
VALUES ('KHNJHN81E30C455Y','Mercoledì','9:00',60,'S1','CT104');
```

3. Testare gli script di creazione e popolamento scritti ai punti precedenti eseguendoli tramite l'interfaccia web di Oracle SQLPLUS.

4. Parte su esecuzione dei comandi di aggiornamento (update, delete, insert)

```
4.1.UPDATE Istruttore
SET Telefono = '+390112333551'
WHERE CodFisc = 'KHNJHN81E30C455Y';
```

```
4.2.UPDATE Programma
SET Sala = 'S4'
WHERE Sala = 'S2';
```

```
4.3.DELETE FROM Corsi
WHERE Corsi.CodC IN (SELECT Programma.CodC
FROM Programma
GROUP BY Programma.CodC
HAVING COUNT(*)=1);
```

Dalla tabella Corsi viene eliminata la riga corrispondente al corso con codice CT101; dalla tabella Programma viene eliminata la riga corrispondente al

corso in questione poiché era stata impostata l'opzione DELETE ON CASCADE in fase di creazione della tabella.

```
4.4.INSERT INTO Corsi (CodC,Nome,Tipo,Livello)
VALUES ('CT120','Spinning estremo','Spinning',10);
```

L'esecuzione del comando genera un errore; infatti la tupla che si tenta di inserire non è ammissibile poiché il valore del Livello non rispetta il vincolo di dominio ( $\text{Livello} \geq 1$  and  $\text{Livello} \leq 4$ ).

```
4.5.DELETE FROM Istruttore
WHERE CodFisc = 'SMTPLA80N31B791Z';
```

Grazie all'opzione DELETE ON CASCADE vengono eliminate anche le righe di Programma contenenti il codice fiscale dell'istruttore che si sta eliminando.