

# ESERCIZI SQL

---



## Esercizio 1

### IMPIEGATO

NOME_BATT	INIZ_INT	COGNOME	<u>SSN</u>	DATA_N	INDIRIZZO	SESSO	STIPENDIO	SUPER_SSN	N_D
-----------	----------	---------	------------	--------	-----------	-------	-----------	-----------	-----

### DIPARTIMENTO

NOME_D	<u>NUMERO_D</u>	SSN_DIR	DATA_INIZIO_DIR
--------	-----------------	---------	-----------------

### SEDI\_DIP

<u>NUMERO_D</u>	<u>SEDE_D</u>
-----------------	---------------

### PROGETTO

NOME_P	<u>NUMERO_P</u>	SEDE_P	NUM_D
--------	-----------------	--------	-------

### LAVORA\_SU

<u>SSN_I</u>	<u>N_P</u>	ORE
--------------	------------	-----

### PERSONA\_A\_CARICO

<u>SSN_I</u>	<u>NOME_PERSONA_A_CARICO</u>	SESSO	DATA_N	PARENTELA
--------------	------------------------------	-------	--------	-----------

# Esercizio 1

## IMPIEGATO

NOME_BATT	INIZ_INT	COGNOME	SSN	DATA_N	INDIRIZZO	SESSO	STIPENDIO	SUPER_SSN	N_D
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1965-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

## DIPARTIMENTO

NOME_D	NUMERO_D	SSN_DIR	DATA_INIZIO_DIR
Ricerca	5	333445555	1988-05-22
Amministrazione	4	987654321	1995-01-01
Sede centrale	1	888665555	1981-06-19

## SEDI\_DIP

NUMERO_D	SEDE_D
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

## LAVORA\_SU

SSN_I	N_P	ORE
123456789	1	32,5
123456789	2	7,5
666884444	3	40,0
453453453	1	20,0
453453453	2	20,0
333445555	2	10,0
333445555	3	10,0
333445555	10	10,0
333445555	20	10,0
999887777	30	30,0
999887777	10	10,0
987987987	10	35,0
987987987	30	5,0
987654321	30	20,0
987654321	20	15,0
888665555	20	NULL

## PROGETTO

NOME_P	NUMERO_P	SEDE_P	NUM_D
ProdottoX	1	Bellaire	5
ProdottoY	2	Sugarland	5
ProdottoZ	3	Houston	5
Informalizzazione	10	Stafford	4
Riorganizzazione	20	Houston	1
Nuova opportunità	30	Stafford	4

## PERSONA\_A\_CARICO

SSN_I	NOME_PERSONA_A_CARICO	SESSO	DATA_N	PARENTELA
333445555	Alice	F	1986-04-05	FIGLIA
333445555	Theodore	M	1983-10-25	FIGLIO
333445555	Joy	F	1968-05-03	CONIUGE
987654321	Abner	M	1942-02-28	CONIUGE
123456789	Michael	M	1988-01-04	FIGLIO
123456789	Alice	F	1988-12-30	FIGLIA
123456789	Elizabeth	F	1967-05-05	CONIUGE

# Esercizio 1

Si specifichino le seguenti interrogazioni SQL sulla base di dati relazionale AZIENDA. Si mostri il risultato di ciascuna interrogazione.

1. Si restituisca il nome di tutti gli impiegati del dipartimento 5 che lavorano più di 10 ore alla settimana sul progetto 'ProdottoX'.
2. Si fornisca un elenco dei nomi di tutti gli impiegati che hanno una persona a carico con il loro stesso nome di battesimo.
3. Si restituisca il nome di tutti gli impiegati che hanno come diretto supervisore 'Franklin Wong'.

## Esercizio 1 - soluzione

1.

```
select nome_batt, cognome
from impiegato, lavora_su, progetto
where n_d = 5 and ssn_i = ssn and ore > 10 and n_p =
numero_p and nome_p = 'ProdottoX'
```

OPPURE

```
select nome_batt, nome_p
from impiegato
      inner join lavora_su on ssn=ssn_l
      inner join progetto on numero_p=n_p
where ore>10 and nome_p='ProdottoX';
```

## Esercizio 1 - soluzione

2.

```
select nome_batt, cognome
from impiegato, persona_a_carico
where ssn_i = ssn and nome_persona_a_carico =
nome_batt
```

OPPURE

```
select nome_batt
from impiegato
      inner join persona_a_carico on ssn=ssn_i
where nome_persona_a_carico=nome_batt;
```

## Esercizio 1 - soluzione

3.

```
select i.nome_batt, i.cognome
from impiegato as i, impiegato as s
where s.ssn = i.super_ssn and s.nome_batt = 'Franklin' and
s.cognome = 'Wong'
```

OPPURE

```
select imp.nome_batt
from impiegato as imp
      inner join impiegato as sup on imp.super_ssn=sup.ssn
where sup.nome_batt='Franklin' and sup.cognome='Wong';
```

## Esercizio 2

STUDENTE	Nome	Numero_studente	Anno_corso	Corso_laurea
	Smith	17	1	CS
	Brown	8	2	CS

INSEGNAMENTO	Nome_insegnamento	Codice_insegnamento	Ore	Dipartimento
	Introduzione all'informatica	CS1310	4	CS
	Struttura dei dati	CS3320	4	CS
	Matematica discreta	MATH2410	3	MATH
	Basi di dati	CS3380	3	CS

MODULO	Identificatore_modulo	Codice_insegnamento	Semestre	Anno	Docente
	85	MATH2410	Autunno	07	King
	92	CS1310	Autunno	07	Anderson
	102	CS3320	Primavera	08	Knuth
	112	MATH2410	Autunno	08	Chang
	119	CS1310	Autunno	08	Anderson
	135	CS3380	Autunno	08	Stone

VOTAZIONE	Numero_studente	Identificatore_modulo	Voto
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PROPEDEUTICITÀ	Codice_insegnamento	Codice_propedeuticità
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

## Esercizio 2

### STUDENTE

Nome	Numero_studente	Anno_corso	Corso_laurea
------	-----------------	------------	--------------

### INSEGNAMENTO

Nome_insegnamento	Codice_insegnamento	Ore	Dipartimento
-------------------	---------------------	-----	--------------

### PROPEDEUTICITÀ

Codice_insegnamento	Codice_propedeuticità
---------------------	-----------------------

### MODULO

Identificatore_modulo	Codice_insegnamento	Semestre	Anno	Docente
-----------------------	---------------------	----------	------	---------

### VOTAZIONE

Numero_studente	Identificatore_modulo	Voto
-----------------	-----------------------	------

## Esercizio 2

Si consideri la base di dati illustrata nelle slide precedenti.

- Quali sono i vincoli di integrità referenziale che dovrebbero valere su questo schema?
- Si definiscano le opportune istruzioni DDL di SQL per definire lo schema della base di dati.

## Esercizio 2 - soluzione

È possibile specificare i seguenti vincoli di integrità referenziale (utilizziamo la notazione  $R.(A_1, \dots, A_n) \rightarrow S.(B_1, \dots, B_n)$  per rappresentare la chiave esterna fra gli attributi  $A_1, \dots, A_n$  di  $R$  (relazione referenziante) e gli attributi  $B_1, \dots, B_n$  di  $S$  (relazione riferita)):

- PROPEDEUTICITA'.(Codice\_Insegnamento)  $\rightarrow$  INSEGNAMENTO.  
(Codice\_Insegnamento)
- PROPEDEUTICITA'.(Codice\_Propedeuticità)  $\rightarrow$  INSEGNAMENTO.  
(Codice\_Insegnamento)
- MODULO.(Codice\_Insegnamento)  $\rightarrow$  INSEGNAMENTO.  
(Codice\_Insegnamento)
- VOTAZIONE.(Numero\_Studente)  $\rightarrow$  STUDENTE.(Numero\_Studente)
- VOTAZIONE.(Identificatore\_Modulo)  $\rightarrow$  MODULO.(Identificatore\_Modulo)

## Esercizio 2 - soluzione

```
CREATE TABLE STUDENTE (
    Nome VARCHAR(30) NOT NULL,
    Numero_Studente INTEGER NOT NULL,
    Anno_Corso CHAR NOT NULL,
    Corso_Laurea CHAR(4),
    PRIMARY KEY (Numero_Studente) );
```

```
CREATE TABLE INSEGNAMENTO (
    Nome_Insegnamento VARCHAR(30) NOT NULL,
    Codice_Insegnamento CHAR(8) NOT NULL,
    Crediti INTEGER,
    Dipartimento CHAR(4),
    PRIMARY KEY (Codice_Insegnamento),
    UNIQUE (Nome_Insegnamento) );
```

## Esercizio 2 - soluzione

```
CREATE TABLE PROPEDEUTICITA' (
    Codice_Insegnamento CHAR(8) NOT NULL,
    Codice_Propedeuticit  CHAR(8) NOT NULL,
    PRIMARY KEY (Codice_Insegnamento, Codice_Propedeuticit ),
    FOREIGN KEY (Codice_Insegnamento)
        REFERENCES INSEGNAMENTO (Codice_Insegnamento),
    FOREIGN KEY (Codice_Propedeuticit )
        REFERENCES INSEGNAMENTO (Codice_Insegnamento) );
```

```
CREATE TABLE MODULO (
    Identificatore_Modulo INTEGER NOT NULL,
    Codice_Insegnamento CHAR(8) NOT NULL,
    Semestre VARCHAR(6) NOT NULL,
    Anno CHAR(4) NOT NULL,
    Docente VARCHAR(15),
    PRIMARY KEY (Identificatore_Modulo),
    FOREIGN KEY (Codice_Insegnamento)
        REFERENCES INSEGNAMENTO (Codice_Insegnamento) );
```

## Esercizio 2 - soluzione

```
CREATE TABLE VOTAZIONE (
    Numero_Studente INTEGER NOT NULL,
    Identificatore_Modulo INTEGER NOT NULL,
    Voto CHAR,
    PRIMARY KEY (Numero_Studente, Identificatore_Modulo),
    FOREIGN KEY (Numero_Studente)
        REFERENCES STUDENTE (Numero_Studente),
    FOREIGN KEY (Identificatore_Modulo)
        REFERENCES MODULO (Identificatore_Modulo) );
```

## Esercizio 3

Si specifichino le seguenti interrogazioni SQL sullo schema della base di dati UNIVERSITA.

Il lettore identifichi quelle che, a suo parere, dovrebbero essere le varie chiavi candidate e esprima i vincoli che devono essere soddisfatti affinché ciascuna chiave candidata sia valida.

1. Si restituisca il nome degli studenti che si laureano in 'CS' (Computer Science, Informatica).
2. Si restituisca il nome degli insegnamenti tenuti dal professor King dal 2007 al 2008.
3. Per ciascun modulo tenuto dal professor King, si restituisca il codice dell'insegnamento, il semestre, l'anno e il numero di studenti che hanno scelto il modulo.
4. Si restituisca il nome e la carriera universitaria di ciascuno studente dell'ultimo anno (Anno\_Corso = 4) laureando in 'CS'. La carriera di uno studente include il nome dell'insegnamento, il codice dell'insegnamento, i crediti, il semestre, l'anno e il voto di ciascun insegnamento completato dallo studente.

## Esercizio 3 - soluzione

```
1. SELECT Nome
FROM STUDENTE
WHERE Corso_Laurea='CS'
```

```
2. SELECT Nome_Insegnamento
FROM INSEGNAMENTO, MODULO
WHERE
INSEGNAMENTO.Codice_Insegnamento=MODULO.Codice_Insegnament
o AND Docente='King' AND (Anno='2007' OR Anno='2008')
```

Un'altra possibile soluzione basata su interrogazioni SQL annidate è la seguente:

```
SELECT Nome_Insegnamento
FROM INSEGNAMENTO
WHERE Codice_Insegnamento IN ( SELECT Codice_Insegnamento FROM
MODULO WHERE Docente='King' AND (Anno='2007' OR Anno='2008') )
```



## Esercizio 3 - soluzione

```
3. SELECT Codice_Insegnamento, Semestre, Anno, COUNT(*)
FROM MODULO, VOTAZIONE
WHERE Docente='King' AND
MODULO.Identificatore_Modulo=VOTAZIONE.Identificatore_
Modulo GROUP BY Codice_Insegnamento, Semestre, Anno
```

```
4. SELECT Nome, Nome_Insegnamento,
I.Codice_Insegnamento, Crediti, Semestre, Anno, Voto
FROM STUDENTE ST, INSEGNAMENTO I, MODULO M,
VOTAZIONE V
WHERE Anno_Corso=4 AND Corso_Laurea='CS' AND
ST.Numero_Studente=V.Numero_Studente AND
V.Identificatore_Modulo=M.Identificatore_Modulo AND
M.Codice_Insegnamento=I.Codice_Insegnamento
```

## Esercizio 4

Si scrivano le istruzioni SQL per eseguire le seguenti operazioni di aggiornamento sullo schema di base di dati UNIVERSITA.

1. Si inserisca un nuovo studente <'Johnson', 25, 1, 'MATH'> nella base di dati.
2. Si modifichi a 2 l'anno di corso dello studente 'Smith'.
3. Si inserisca un nuovo insegnamento <'Ingegneria della conoscenza', 'CS4390', 3, 'CS'>.
4. Si elimini il record dello studente il cui nome è 'Smith' e il cui numero è 17.

## Esercizio 4 - soluzione

1. INSERT INTO STUDENTE VALUES ('Johnson', 25, 1, 'MATH')
2. UPDATE STUDENTE SET Anno\_Corso = 2 WHERE Nome='Smith'
3. INSERT INTO INSEGNAMENTO VALUES (Ingegneria della Conoscenza','CS4390', 3,'CS')
4. DELETE FROM STUDENTE WHERE Nome='Smith' AND Numero\_Studente=17

## Esercizio 5

Si specifichino le seguenti interrogazioni sulla base di dati AZIENDA. Si mostrino i risultati.

1. Si trovino i nomi dei dipartimenti e i numeri degli impiegati che lavorano in ogni dipartimento i cui impiegati hanno uno stipendio medio di più di 30.000 dollari.
2. Si supponga di voler individuare il numero di impiegati maschi in ciascun dipartimento anziché tutti i dipendenti. Si può specificare questa interrogazione in SQL? Perché? O perché no?

## Esercizio 6

Si specifichino le seguenti interrogazioni SQL sullo schema della base di dati UNIVERSITA.

1. Si trovino i nomi e i dipartimenti di specializzazione degli studenti più meritevoli (studenti che hanno conseguito il massimo dei voti in tutti loro insegnamenti).
2. Si trovino i nomi e i dipartimenti di specializzazione di tutti gli studenti che non hanno il massimo dei voti in tutti gli insegnamenti da loro frequentati.

## Esercizio 7

Si specifichino in SQL le seguenti interrogazioni sulla base di dati AZIENDA, utilizzando i concetti illustrati in questo capitolo e in particolare quello di interrogazione nidificata.

1. Si restituisca il cognome degli impiegati che lavorano nello stesso dipartimento dell'impiegato che percepisce lo stipendio più elevato.
2. Si restituisca il cognome degli impiegati il cui supervisore ha un supervisore che possiede un SSN con valore '888665555'.
3. Si restituisca il cognome degli impiegati che percepiscono oltre \$10.000 in più dell'impiegato con lo stipendio più basso.

## Esercizio 7 - soluzione

1. SELECT COGNOME FROM IMPIEGATO WHERE N\_D =  
( SELECT N\_D FROM IMPIEGATO WHERE STIPENDIO =  
( SELECT MAX(STIPENDIO) FROM IMPIEGATO ) )

2. SELECT COGNOME FROM IMPIEGATO WHERE  
SUPER\_SSN IN  
( SELECT SSN FROM IMPIEGATO WHERE SUPER\_SSN =  
'888665555' )

3. SELECT COGNOME FROM IMPIEGATO WHERE  
STIPENDIO >= 10000 +  
( SELECT MIN(STIPENDIO) FROM IMPIEGATO )