

## CREAZIONE STRUTTURA DI UNA TABELLA

by Marinsalta Fausto

```
CREATE TABLE Personale
  Matricola char(5),
  Cognome char(30),
  Nome char(20),
  Codfis char(16) not null,
  Assunzione date,
  Filiale smallint,
  Funzione char(15),
  Livello smallint,
  Stipbase integer,
  Via char(25),
  Cap char(5),
  Citta char(20),
  Provincia char(2);
```

\*\*\*\*\*

*MODIFICA STRUTTURA DI UNA TABELLA (inserimento di un nuovo campo)*

```
ALTER TABLE Personale
  ADD Nascita date;
```

*MODIFICA STRUTTURA DI UNA TABELLA(cancellazione di un campo)*

```
ALTER TABLE Personale
  DROP Nascita;
```

*CREAZIONE DI UN INDICE IN UNA TABELLA*

```
CREATE UNIQUE INDEX Index1
  ON Personale (cognome, nome);
```

*CANCELLAZIONE DI UNA TABELLA*

```
DROP TABLE Personale;
```

*INSERIMENTO DI VALORI (nuova riga) IN UNA TABELLA*

```
INSERT INTO Personale
  (Matricola, Cognome, Nome, Codfis, Nascita, Assunzione, Filiale,
  Funzione, Livello, Stipbase, Via, Cap, Citta, Provincia)
VALUES ('abcde', 'cassa', 'mario', 'cssrrt53a14a2710', '08/10/1980', '08/12/1990', 3,
'dirigente' , 5, 888, 'via esino 44', '60100', 'ancona', 'an');
```

*MODIFICA DI UN CAMPO IN UNA UNA O PIU' RIGHE DI UNA TABELLA*

```
UPDATE Personale
  SET livello = 6
  WHERE Matricola = 'abcde';
```

**UPDATE Personale**  
**SET Stipbase = Stipbase \* 1,05**  
**WHERE livello > 6;**

***CANCELLAZIONE DI UNA O PIU' RIGHE IN UNA TABELLA***

**DELETE FROM Personale**  
**WHERE Matricola = 'abcde';**

**DELETE FROM Personale**  
**WHERE Stipbase < 750;**

***INTERROGAZIONE DI UNA BASE DATI***

**SELECT Cognome, Nome, Codfisc**  
**FROM Personale**  
**WHERE Funzione = 'impiegato';**

**SELECT \***  
**FROM Personale**  
**WHERE Provincia = 'an';**  
*\*\*\*\*\* (all e' sottointeso) \*\*\*\*\**

**SELECT all \***  
**FROM Personale**  
**WHERE Provincia = 'an';**  
*\*\*\*\*\* (all puo' essere omesso) \*\*\*\*\**

**SELECT DISTINCT Funzione**  
**FROM Personale;**  
*\*\*\*\*(viene prodotto un elenco delle funzioni Distinct elimina le righe duplicate)\*\*\*\**

**SELECT DISTINCT Provincia AS PV**  
**FROM Personale;**  
*\*\*\*\*(viene sostituita l'intestazione di default provincia con PV) \*\*\*\**

**SELECT Cognome, Nome, Stipbase AS Attuale, Stipbase \* 1,5 AS Nuovo**  
**FROM Personale;**  
*\*\*\*\* l'intestazione Stipbase e' sostituita con Attuale; viene creata una colonna nuova intestata a Nuovo e calcolato il valore corrispondente \*\*\*\**

**SELECT Cognome, Nome, Codfisc**  
**FROM Personale**  
**WHERE Funzione = 'dirigente';**  
*\*\*\* estrae solo i dipendenti aventi la funzione di dirigente \*\*\**

**SELECT Cognome, Nome, Codfisc**  
**FROM Personale**  
**WHERE Funzione = (tipoFunzione);**  
*\*\*\* in fase di avvio della select viene chiesto di inizializzare la variabile tipoFunzione dopo di che vengono estratti solo i dipendenti aventi come funzione il valore digitato \*\*\**

## ***DEFINIZIONE DELLE RELAZIONI TRA TABELLE***

### ***CREAZIONE STRUTTURA TABELLA (Personale)***

```
CREATE TABLE Personale
  Matricola    char(5),
  Cognome     char(30),
  Nome        char(20),
  Codfis      char(16) not null,
  Assunzione  date,
  Filiale     smallint,
  Funzione    char(15),
  Livello     smallint,
  Stipbase    integer,
  Via         char(25),
  Cap         char(5),
  Citta       char(20),
  Provincia  char(2),
  Dirigente  char(5); ← identifica il codice del dirigente corrispondente
```

### ***CREAZIONE STRUTTURA TABELLA (Dipendenza)***

```
CREATE TABLE Dipendenza
  Codfil      smallint,
  Descrizione char(30),
  Indirizzo   char(20) ,
  Via         char(25) ;
```

\*\*\*\*\*

### ***CONGIUNZIONE TRA DUE TABELLE DISTINTE Personale e Dipendenza***

```
SELECT *
FROM Personale, Dipendenza
WHERE Personale.Filiale = Dipendenza.Codfil;
```

*\*\*\*trattasi di congiunzione equi-join (inner-join) ovvero vengono combinate solo le righe per le quali si possono trovare valori uguali negli attributi che si corrispondono\*\*\**

### ***CONGIUNZIONE TRA LA TABELLA Personale e se stessa (utilizzo di alias)***

```
SELECT Tab1.Cognome, Tab1.Nome, Tab2.Cognome
FROM Personale AS Tab1, Personale AS tab2
WHERE Tab1.Dirigente = Tab2.Matricola;
```

*\*\*\*trattasi di congiunzione self-join su una tabella (per ciascun dipendente si ottiene il cognome, il nome, ed il nome del corrispondente dirigente). \*\*\**

### ***CONGIUNZIONE TRA TABELLE E CONDIZIONI MULTIPLE***

```
SELECT Cognome, Nome, Descrizione, Indirizzo
FROM Personale, Dipendenza
WHERE Personale.Filiale = Dipendenza.Codfil
AND Personale.funzione = 'impiegato';
```

*\*\*\*viene effettuata una prima selezione sulla base della uguaglianza tra FILIALE e CODFIL, quindi vengono scartati tutti i dipendenti per i quali funzione <> 'impiegato' \*\*\**

**CONGIUNZIONE TRA TABELLE E CONDIZIONI MULTIPLE (segue)**

**SELECT** Personale.Cognome, personale.Nome, Dipendenza.Descrizione,  
Dipendenza.Indirizzo

**FROM** Personale, Dipendenza

**WHERE** Personale.Filiale = Dipendenza.Codfil

**AND** Personale.funzione = 'impiegato';

*\*\*\*viene effettuata una prima selezione sulla base della uguaglianza tra FILIALE e CODFIL, quindi vengono scartati tutti i dipendenti per i quali funzione <> 'impiegato' \*\*\**

**CREAZIONE TRAMITE SELECT DI UNA NUOVA TABELLA DA UNA TABELLA  
ORIGINARIA (clausola INTO)**

**SELECT \* INTO** Manager

**FROM** Personale

**WHERE** funzione = 'impiegato';

*\*\*\*viene creata una nuova tabella Manager contenete tutti i campi della tabella Personale ed istanziata con le sole righe per cui funzione = 'impiegato' \*\*\**

**TRASFERIMENTO DI RIGHE DA UNA TABELLA ED ACCODAMENTO SU UN'ALTRA  
MEDIANTE SELECT (clausola INSERT INTO)**

**INSERT INTO** Personale

**SELECT \***

**FROM** NuoviAssunti;

*\*\*\*vengono trasferiti i righe dalla tabella NuoviAssunti sulla tabella Personale \*\*\**

\*\*\*\*\*

**FUNZIONE DI AGGREGAZIONE (COUNT)**

**SELECT COUNT (\*)**

**FROM** Personale;

*\*\*\*vengono contate tutte le righe della tabella Personale (righe nulle incluse) \*\*\**

**SELECT COUNT (livello)**

**FROM** Personale;

*\*\*\*vengono contate tutte le righe della tabella Personale per le quali e' specificato il livello (ovvero contiene un valore non nullo) \*\*\**

**SELECT COUNT (\*)**

**FROM** Personale

**WHERE** Provincia = 'an';

*\*\*\*vengono contate tutte le righe della tabella Personale per le quali la Provincia vale 'an'\*\*\**

**SELECT COUNT (DISTINCT Livello)**

**FROM** Personale

**WHERE** Provincia = 'an';

*\*\*\*vengono contate tutte le righe della tabella Personale per le quali la Provincia vale 'an' con soppressione dal conteggio di tutte le righe aventi il campo livello duplicato, ovvero vengono contati quanti livelli esistono per i dipendenti della provincia di 'an') \*\*\**

**SELECT COUNT (DISTINCT Livello) AS Livelli**

**FROM Personale**  
**WHERE Provincia = 'an';**

*\*\*\*vengono contate tutte le righe della tabella Personale per le quali la Provincia vale 'an' con soppressione dal conteggio di tutte le righe aventi il campo livello duplicato, ovvero vengono contati quanti livelli esistono per i dipendenti della provincia di 'an' ed introdotta l'intestazione 'livelli'\*\*\**

**FUNZIONE DI AGGREGAZIONE (SUM)**

**CREATE TABLE Fatture**

**NumFat           smallint,**  
**PrezUni           integer,**  
**Qta               integer;**

\*\*\*\*\*

**SELECT SUM (Stipbase)**

**FROM Personale**  
**WHERE Livello = 5;**

*\*\*\*viene totalizzato il campo Stipbase della tabella Personale per le quali il livello vale 5\*\*\**

**SELECT SUM (PrezUnit \* Qta) AS Totale**

**FROM Fatture;**  
**WHERE Provincia = 'an';**

*\*\*\* viene calcolato per ogni riga della tabella Fatture il prezzo totale come campo calcolato (PrezUnit \* Qta) e totalizzato il risultato in Totale\*\*\**

**FUNZIONE DI AGGREGAZIONE (AVG)**

**SELECT AVG (Stipbase)**

**FROM Personale**  
**WHERE Funzione = 'impiegato';**

*\*\*\*viene calcolato per tutti gli impiegati presenti nella tabella Personale lo Stipendio medio pagato inteso come sommatoria dello stipendio base degli impiegati / numero di impiegati.\*\*\**

**FUNZIONE DI AGGREGAZIONE (MIN e MAX)**

**SELECT MIN (Stipbase), MAX (Stipbase)**

**FROM Personale;**

*\*\*\*vengono ricercati lo stipendio minimo e massimo percepito dai dipendenti contenuti nella tabella Personale\*\*\**

**SELECT MIN (Cognome), MAX (Cognome)**

**FROM Personale;**

*\*\*\*vengono restituito il primo cognome e l'ultimo cognome (in senso alfabetico) dei dipendenti contenuti nella tabella Personale\*\*\**

**ORDINAMENTI (order) e RAGGRUPPAMENTI (group)**

**SELECT Cognome, Nome, Codfisc, Nascita**

**FROM Personale**  
**ORDER BY Cognome, Nome;**

\*\*\* vengono visualizzati i dipendenti nei campi (Cognome, Nome, Codfisc, Nascita) ordinati per Cognome e a pari Cognome per Nome \*\*\*

\*\*\*\*\*

```
SELECT Livello, COUNT (Livello) AS Conteggio
FROM Personale
WHERE Funzione = 'impiegato'
GROUP BY Livello;
```

\*\*\*viene fornito l'elenco dei livelli esistenti tra i dipendenti che svolgono la funzione di Impiegato con il numero di dipendenti per ciascun livello \*\*\*

```
SELECT Funzione, AVG (Stipbase)
FROM Personale
GROUP BY Funzione
HAVING COUNT (*) > 2;
```

\*\*\*viene fornita la lista delle funzioni dei dipendenti ed indicato lo stipendio medio per ciascuna funzione dopo aver raggruppato i dipendenti per funzione purché i dipendenti con quella funzione siano più di 2 \*\*\*

```
SELECT Filiale, COUNT (Filiale) AS Conteggio
FROM Personale
WHERE Funzione = 'impiegato'
GROUP BY Filiale
HAVING COUNT (*) > 10;
```

\*\*\*viene fornito l'elenco delle filiali nelle quali ci sono più di 10 impiegati \*\*\*

\*\*\* N.B. La clausola WHERE effettua il controllo condizionale sulla righe della tabella, la clausola HAVING effettua il controllo condizionale sui risultati delle funzioni di aggregazione applicate a gruppi di righe. \*\*\*

### CONDIZIONI DI RICERCA (BETWEEN, IN, LIKE, IS NULL)

**BETWEEN** ➔ Controlla se un valore è compreso all'interno di un intervallo di valori, inclusi gli estremi.

**IN** ➔ Controlla se un valore appartiene ad un insieme specificato di valori.

**LIKE** ➔ Confronta il valore di un attributo di tipo carattere con un modello di stringa che può contenere caratteri jolly (\_), (%).

```
SELECT Cognome, Nome, Funzione
FROM Personale
WHERE Assunto BETWEEN 01/01/95 and 12/31/98;
```

\*\*\*viene fornito l'elenco dei dipendenti nei campi (Cognome, Nome, Funzione) che sono stati assunti tra il 01/01/95 ed il 12/31/98 \*\*\*

```
SELECT *
FROM Personale
WHERE Provincia IN ('mi', 'an', 'bs');
```

\*\*\*viene fornito l'elenco dei dipendenti con indicato tutti i campi per i quali il campo Provincia vale o 'mi' o 'an' o 'bs' \*\*\*

**SELECT \***

**FROM Personale**

**WHERE Provincia NOT IN ('mi', 'an', 'bs');**

*\*\*\*viene fornito l'elenco dei dipendenti con indicato tutti i campi per i quali il campo Provincia non assume il valore di 'mi', 'an', 'bs' \*\*\**

**SELECT Cognome, Filiale**

**FROM Personale**

**WHERE Cognome LIKE 'Ros%';**

*\*\*\*viene fornito l'elenco dei dipendenti nei campi (Cognome, Filiale) per i quali il campo cognome inizia per 'Ros' \*\*\**

**SELECT Cognome, Filiale**

**FROM Personale**

**WHERE Cognome NOT LIKE 'Ros%';**

*\*\*\*viene fornito l'elenco dei dipendenti nei campi (Cognome, Filiale) per i quali il campo cognome non inizia per 'Ros' \*\*\**

**SELECT Cognome, Nome**

**FROM Personale**

**WHERE Provincia IS NOT NULL;**

*\*\*\*viene fornito l'elenco dei dipendenti nei campi (Cognome, Nome) per i quali il campo Provincia non e' un valore nullo ovvero non ci e' stato inserito nulla \*\*\**

## **COMANDI PER LA SICUREZZA E CONTROLLO DEGLI ACCESSI ALLE TABELLE)**

### **GRANT UPDATE**

**ON Personale**

**TO User1, User2;**

*\*\*\*viene concesso il diritto di modificare la tabella Personale agli utenti User1 ed User2 \*\*\**

### **REVOKE UPDATE**

**ON Personale**

**FROM User1, User2;**

*\*\*\*viene revocato il permesso di accesso a modificare la tabella Personale agli utenti User1 ed User2 \*\*\**

### **GRANT UPDATE (Livello, Stipbase)**

**ON Personale**

**TO User3;**

*\*\*\*viene concesso il diritto di modificare i Campi livello e Stipbase della tabella Personale all'utente User3 \*\*\**

### **\*\*\*\* GRANT e REVOKE operano con le seguenti parole chiave:**

**ALTER** → per aggiungere e/o modificare colonne oppure per modificare i tipi di dati

**DELETE** → per eliminare righe dalle tabelle

**INDEX** → per creare indici

**INSERT** → per inserire nuove righe nelle tabelle

**SELECT** → per ricercare i dati dalle tabelle

**UPDATE** → per cambiare i valori contenuti nelle tabelle

**ALL** → per specificare tutti i permessi precedenti

**\*\*\*\* GRANT e REVOKE unitamente alle parole chiave SELECT ed UPDATE possono operare anche su specifici campi della tabella. I campi su cui operare vengono indicati all'interno di parentesi tonde e separati tra di loro dalla virgola \*\*\*\***

## ***VISTE LOGICHE***

**CREATE VIEW Impiegati**

**AS SELECT \***

**FROM Personale**

**WHERE Funzione = 'Impiegato';**

*\*\*\*viene creata una vista di nome Impiegati contenente i dati dei dipendenti con funzione di impiegato \*\*\**

**DROP VIEW Impiegati;**

*\*\*\*viene cancellata la vista di nome Impiegati \*\*\**

**GRANT SELECT**

**ON Impiegati**

**TO User3;**

*\*\*\*viene concesso all'utente di nome User3 di modificare i dati contenuti nella vista di nome Impiegati \*\*\**

## ***INTEGRITA' DEI DATI***

**LOCK TABLE** ➔ *Limita l'accesso di determinati utenti alla tabella*

**UNLOCK TABLE** ➔ *Ripristina l'accesso di determinati utenti alla tabella*

**RECOVER TABLE** ➔ *Consente il recupero di una tabella da una copia di sicurezza del DB*

**CHECK TABLE** ➔ *Controlla la corrispondenza dei dati di una tabella con i suoi indici*

**REPAIR TABLE** ➔ *Ricostruisce gli indici di una tabella*

## GESTIONE DELLE TRANSAZIONI

**BEGIN TRANSACTION** ➔ Consente la manipolazione dei dati di una tabella ad uno specifico utente e nel contesto inibisce la manipolazione degli stessi ad altri utenti ai quali permane la possibilità di visualizzazione. (Utilissimo quando si deve operare in sequenza su più tabelle distinte e nel contesto si vuol assicurare l'integrità dei dati dell'intero sistema da cadute di tensioni e/o da operazioni non andate a buon fine)

**COMMIT** ➔ Rende effettive le manipolazioni effettuate sui dati dopo il comando: **BEGIN TRANSACTION**.

**ROLLBACK** ➔ Ripristina il data base alla situazione iniziale ignorando tutte le manipolazioni effettuate sui dati dopo il comando: **BEGIN TRANSACTION**.

## INTERROGAZIONI NIDIFICATE (subquery)

```
SELECT Cognome, Nome
FROM Personale
WHERE Stipbase <
      (SELECT AVG (Stipbase)
       FROM Personale) ;
```

\*\*\*viene fornito l'elenco con cognome e nome dei dipendenti che hanno lo stipendio base inferiore allo stipendio medio di tutti i dipendenti \*\*\*

```
SELECT Cognome, Nome, Descrizione
FROM Personale, Dipendenza
WHERE Filiale = Codfil
      AND Stipbase = (SELECT MAX (Stipbase)
                     FROM Personale
                     WHERE Funzione = 'impiegato')
```

**ORDER BY** Cognome, Nome;

\*\*\*viene fornito l'elenco dei dipendenti con cognome e nome e con la descrizione della filiale di lavoro per i quali lo stipendio base risulta al valore massimo tra tutti gli stipendi dei dipendenti con la funzione di impiegato. L'elenco viene prodotto ordinato per cognome del dipendente e a pari cognome per nome. \*\*\*

## PREDICATO (ANY)

```
SELECT Cognome, Nome, Funzione
FROM Personale
WHERE Funzione <> 'impiegato'
      AND Stipbase > ANY (SELECT Stipbase
                          FROM Personale
                          WHERE Funzione = 'impiegato') ;
```

\*\*\*viene fornito l'elenco dei dipendenti con cognome e nome e con la descrizione della funzione svolta che non sono impiegati e che hanno lo stipendio base superiore a quello di uno qualsiasi tra gli impiegati. \*\*\*

*PREDICATO (ALL)*

```
SELECT Cognome, Nome, Funzione  
FROM Personale  
WHERE Funzione <> 'impiegato'  
AND Stipbase > ALL (SELECT Stipbase  
FROM Personale  
WHERE Funzione = 'impiegato') ;
```

*\*\*\*viene fornito l'elenco dei dipendenti con cognome e nome e con la descrizione della funzione svolta che non sono impiegati e che hanno lo stipendio base superiore a quello di tutti gli impiegati \*\*\**

*PREDICATO (IN)*

```
SELECT Cognome, Nome  
FROM Personale  
WHERE Filiale IN (SELECT Stipbase  
FROM Personale  
GROUP BY Filiale  
HAVING COUNT (*) > 10) ;
```

*\*\*\*viene fornito l'elenco dei dipendenti con cognome e nome che lavorano nelle filiali che hanno piu' di 10 dipendenti. \*\*\**

*PREDICATO (EXISTS)*

```
SELECT Cognome, Nome  
FROM Personale  
WHERE exists (SELECT *  
FROM Personale  
WHERE Livello = 6) ;
```

*\*\*\*viene fornito l'elenco dei dipendenti con cognome e nome solo se esistono dipendenti di sesto livello \*\*\**

**ESEMPIO (01)**

**Tabella: Istituti**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodiceIstituto</i>	<i>Primaria</i>	<i>Carattere</i>	<i>4</i>
<i>Denominazione</i>		<i>Carattere</i>	<i>35</i>
<i>Indirizzo</i>		<i>Carattere</i>	<i>25</i>
<i>Telefono</i>		<i>Carattere</i>	<i>12</i>

**Tabella: Studenti**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodiceStud</i>	<i>Primaria</i>	<i>Numerico</i>	<i>5</i>
<i>CognStud</i>		<i>Carattere</i>	<i>30</i>
<i>NomStud</i>		<i>Carattere</i>	<i>20</i>
<i>DataNascita</i>		<i>Data/Ora</i>	<i>8</i>
<i>Classe</i>		<i>Numerico</i>	<i>1</i>
<i>CodiceIstituto</i>	<i>Esterna</i>	<i>Carattere</i>	<i>4</i>

**Tabella: Manifestazioni**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodiceManif</i>	<i>Primaria</i>	<i>Carattere</i>	<i>3</i>
<i>Descrizione</i>		<i>Carattere</i>	<i>25</i>
<i>Luogo</i>		<i>Carattere</i>	<i>20</i>
<i>DataInizio</i>		<i>Data/Ora</i>	<i>8</i>

**Tabella: Professori**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodiceProf</i>	<i>Primaria</i>	<i>Numerico</i>	<i>3</i>
<i>CognProf</i>		<i>Carattere</i>	<i>30</i>
<i>NomProf</i>		<i>Carattere</i>	<i>20</i>
<i>CodiceManif</i>	<i>Esterna</i>	<i>Carattere</i>	<i>4</i>
<i>CodiceIstituto</i>	<i>Esterna</i>	<i>Carattere</i>	<i>3</i>

**Tabella: Iscrizioni**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodiceStud</i>	<i>Esterna</i>	<i>Numerico</i>	<i>5</i>
<i>CodiceManif</i>	<i>Esterna</i>	<i>Carattere</i>	<i>3</i>
<i>DataIscrizione</i>		<i>Data/Ora</i>	<i>8</i>

**SELECT COUNT (\*)**

**FROM Manifestazioni, Iscrizioni**

**WHERE Manifestazioni.CodiceManif = Iscrizioni.CodiceManif**

**AND Descrizione = [manifestazione da controllare];**

\*\*\*viene restituito il numero degli studenti che partecipano ad una determinata manifestazione sportiva appositamente digitata \*\*\*

```
SELECT CognProf, NomeProf  
FROM Manifestazioni, Professori  
WHERE Manifestazioni.CodiceManif = Professori.CodiceManif  
AND Descrizione = [manifestazione da controllare];
```

*\*\*\*viene restituito l'elenco anagrafico degli allenatori (professori) di una determinata manifestazione sportiva appositamente digitata \*\*\**

```
SELECT Denominazione, COUNT (DISTINCT CodiceStud)  
FROM Stidenti, Istituti  
WHERE Studenti.CodiceIstituto = Istituti.CodiceIstituto  
GROUP BY Denominazione;
```

*\*\*\*viene restituito l'elenco delle scuole (Denominazione) con indicato il numero degli studenti che vi partecipano \*\*\**

```
SELECT Denominazione, Indirizzo, Telefono  
FROM Manifestazioni, Iscrizioni, Studenti, Istituti  
WHERE Manifestazioni.CodiceManif = Iscrizioni.CodiceManif  
AND Studenti.CodiceStud = Iscrizioni.CodiceStud  
AND Studenti.CodiceIstituto = Istituti.CodiceIstituto  
AND Descrizione = [manifestazione da controllare]
```

```
GROUP BY Denominazione, Indirizzo, Telefono  
HAVING COUNT (*) > 0;
```

*\*\*\*viene restituito l'elenco delle scuole (denominazione, indirizzo, telefono) che hanno almeno uno studente che partecipa ad una determinata manifestazione sportiva appositamente digitata \*\*\**

```
SELECT CognProf, NomeProf, Denominazione  
FROM Professori, Istituti  
WHERE Professori.CodiceIstituto = Istituti.CodiceIstituto  
ORDER BY CognProf, NomeProf;
```

*\*\*\*viene restituito l'elenco degli allenatori (cognome, nome) con la corrispondente denominazione della scuola di appartenenza il tutto ordinato per cognome e nome dell'allenatore (professore). \*\*\**

```
SELECT Descrizione, COUNT (*)  
FROM Manifestazioni, Iscrizioni, Studenti, Istituti  
WHERE Manifestazioni.CodiceManif = Iscrizioni.CodiceManif  
AND Studenti.CodiceStud = Iscrizioni.CodiceStud  
AND Studenti.CodiceIstituto = Istituti.CodiceIstituto  
AND Descrizione = [scuola da controllare]
```

```
GROUP BY Descrizione;
```

*\*\*\*viene restituito il numero di studenti di una determinata scuola da controllare che partecipano a ciascuna delle manifestazioni sportive. \*\*\**

## ESEMPIO (02)

**Tabella: Rappresentanti**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodiceRap</i>	<i>Primaria</i>	<i>Carattere</i>	<i>4</i>
<i>CognomeRap</i>		<i>Carattere</i>	<i>30</i>
<i>NomeRap</i>		<i>Carattere</i>	<i>20</i>
<i>Zona</i>		<i>Numerico</i>	<i>4</i>

**Tabella: Clienti**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodiceCli</i>	<i>Primaria</i>	<i>Carattere</i>	<i>5</i>
<i>RagioneSociale</i>		<i>Carattere</i>	<i>40</i>
<i>Indirizzo</i>		<i>Carattere</i>	<i>25</i>
<i>Telefono</i>		<i>Carattere</i>	<i>12</i>
<i>PartitaIva</i>		<i>Carattere</i>	<i>11</i>
<i>CodiceRap</i>	<i>Esterna</i>	<i>Carattere</i>	<i>4</i>

**Tabella: Fatture**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>NumeroFatt</i>	<i>Primaria</i>	<i>Numerico</i>	<i>5</i>
<i>DataFatt</i>		<i>Data/Ora</i>	<i>8</i>
<i>Importo</i>		<i>Numerico</i>	<i>12</i>
<i>CodiceCli</i>		<i>Carattere</i>	<i>8</i>

```
SELECT RagioneSociale, Indirizzo, Telefono, PartitaIva
FROM Manifestazioni, Iscrizioni
WHERE Clienti.CodiceRap = Rappresentanti.CodiceRap
AND CognomeRap = [rappresentante da controllare];
```

\*\*\*viene restituito l'elenco dei clienti di uno specifico rappresentante selezionato (digitato) \*\*\*

```
SELECT COUNT (*)
FROM Clienti, Rappresentanti
WHERE Clienti.CodiceRap = Rappresentanti.CodiceRap
AND CognomeRap = [rappresentante da controllare];
```

\*\*\*viene restituito il numero dei clienti di uno specifico rappresentante selezionato (digitato) \*\*\*

```
SELECT CognomeRap, NomeRap
FROM Clienti, Rappresentanti
WHERE Clienti.CodiceRap = Rappresentanti.CodiceRap
AND RagioneSociale = [Ragione sociale cliente da controllare];
```

\*\*\*viene restituito il cognome e nome del rappresentante di un determinato cliente selezionato (digitato) \*\*\*

```
SELECT *  
FROM Fatture  
WHERE CodiceCli = [codice cliente da controllare]  
ORDER BY DataFatt;  
***viene restituito l'elenco delle fatture di di un determinato cliente selezionato (digitato) ordinate per data ***
```

```
SELECT CodiceRap, SUM [Importo]  
FROM Fatture, Clienti  
WHERE Fatture.CodiceCli = Clienti.CodiceCli  
GROUP BY CodiceRap;  
***viene restituito il fatturato raggruppato per codice rappresentante ***
```

```
SELECT CodiceRap, COUNT (*)  
FROM Fatture, Clienti  
WHERE Fatture.CodiceCli = Clienti.CodiceCli  
GROUP BY CodiceRap;  
***viene restituito il numero delle fatture emesse raggruppate per codice rappresentante ***
```

```
SELECT RagioneSociale, DataFatt, Importo  
FROM Fatture, Clienti  
WHERE Fatture.CodiceCli = Clienti.CodiceCli  
AND Importo = (SELECT MAX (Importo)  
FROM Fatture);  
***viene restituito il nome del Cliente e la data di emissione della fattura relativa al cliente in questione avente l'importo massimo ***
```

```
SELECT CognomeRap, NomeRap  
FROM Fatture, Rappresentanti, Clienti  
WHERE Clienti.CodiceRap = Rappresentanti.CodiceRap  
AND Fatture.CodiceCli = Clienti.CodiceCli  
AND Importo = (SELECT MAX (Importo)  
FROM Fatture);  
***viene restituito il Cognome e nome del rappresentante corrispondente alla fattura con importo massimo ***
```

```
SELECT CodiceCli, RagioneSociale  
FROM Fatture, Clienti  
WHERE Fatture.CodiceCli = Clienti.CodiceCli  
GROUP BY CodiceCli, RagioneSociale  
HAVING SUM (Importo) > [cifra prefissata];  
***viene restituito il codice e la RagioneSociale dei clienti per i quali il fatturato totale e' superiore ad una cifra prefissata (digitata) ***
```

```
SELECT CodiceRap, CognomeRap, NomeRap  
FROM Fatture, Rappresentanti, Clienti  
WHERE Clienti.CodiceRap = Rappresentanti.CodiceRap  
AND Fatture.CodiceCli = Clienti.CodiceCli  
GROUP BY CodiceRap, CognomeRap, NomeRap  
HAVING COUNT (*) > [numero prefissato];
```

*\*\*\*viene restituito il codice, cognome e nome dei rappresentanti per i quali il numero delle fatture emesse e' superiore ad un certo numero prefissato (digitato) \*\*\**

**ESEMPIO (03)**

**Tabella: Nazioni**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodiceNaz</i>	<i>Primaria</i>	<i>Carattere</i>	<i>3</i>
<i>Denominazione</i>		<i>Carattere</i>	<i>20</i>
<i>Moneta</i>		<i>Carattere</i>	<i>15</i>
<i>Clima</i>		<i>Carattere</i>	<i>30</i>

**Tabella: Clienti**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodiceCli</i>	<i>Primaria</i>	<i>Numerico</i>	<i>4</i>
<i>Cognome</i>		<i>Carattere</i>	<i>30</i>
<i>Nome</i>		<i>Carattere</i>	<i>20</i>
<i>Indirizzo</i>		<i>Carattere</i>	<i>25</i>
<i>Telefono</i>		<i>Carattere</i>	<i>12</i>

**Tabella: Organizzazioni**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodiceOrg</i>	<i>Primaria</i>	<i>Carattere</i>	<i>4</i>
<i>NomeOrg</i>		<i>Carattere</i>	<i>30</i>
<i>TelefOrg</i>		<i>Carattere</i>	<i>12</i>

**Tabella: Pacchetti**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodicePac</i>	<i>Primaria</i>	<i>Carattere</i>	<i>5</i>
<i>Descrizione</i>		<i>Carattere</i>	<i>50</i>
<i>Modalita</i>		<i>Carattere</i>	<i>40</i>
<i>Prezzo</i>		<i>Numerico</i>	<i>12</i>
<i>CodiceNaz</i>	<i>Esterna</i>	<i>Carattere</i>	<i>3</i>
<i>CodiceOrg</i>	<i>Esterna</i>	<i>Carattere</i>	<i>4</i>

**Tabella: Acquisti**

<b>CAMPO</b>	<b>CHIAVE</b>	<b>FORMATO</b>	<b>DIMENSIONE</b>
<i>CodiceCli</i>	<i>Esterna</i>	<i>Numerico</i>	<i>4</i>
<i>CodicePac</i>	<i>Esterna</i>	<i>Carattere</i>	<i>5</i>
<i>DataAcquisto</i>		<i>Data/Ora</i>	<i>8</i>

**SELECT \***

**FROM Pacchetti**

**WHERE Prezzo < [prezzo massimo];**

\*\*\*viene restituito l'elenco dei viaggi vacanza con prezzo inferiore ad un prezzo prefissato (digitato) \*\*\*

**SELECT Descrizione, Modalita, Prezzo**

**FROM Nazioni, Pacchetti**

**WHERE Nazioni.CodiceNaz = Pacchetti.CodiceNaz  
AND Denominazione = [localita' da controllare];**

*\*\*\*viene restituito l'elenco dei soggiorni con prezzo relativi ad una specifica localita' selezionata (digitata) \*\*\**

**SELECT Cognome, Nome, Indirizzo  
FROM Nazioni, Pacchetti, Acquisti, CodiceNaz  
WHERE Nazioni.CodiceNaz = Pacchetti.CodiceNaz  
AND Pacchetti.CodicePac = Acquisti.CodicePac  
AND Acquisti.CodiceCli = Clienti.CodiceCli  
AND Denominazione = [localita' da controllare];**

*\*\*\*viene restituito l'elenco dei clienti con cognome nome ed indirizzo che hanno fatto un viaggio con una determinata destinazione selezionata (digitata) \*\*\**

**SELECT COUNT (\*)  
FROM Pacchetti, Organizzazioni  
WHERE Pacchetti.CodiceOrg = Organizzazioni.CodiceOrg  
AND NomeOrg = [organizzazione da controllare];**

*\*\*\*viene restituito il numero dei pacchetti offerti da una organizzazione turistica prefissata (valore digitato) \*\*\**

**SELECT NomeOrg, TelefOrg  
FROM Pacchetti, Organizzazioni  
WHERE Pacchetti.CodiceOrg = Organizzazioni.CodiceOrg  
GROUP BY NomeOrg, TelefOrg  
HAVING COUNT (\*) > 4;**

*\*\*\*viene restituito l'elenco delle organizzazioni turistiche con nome e telefono che hanno almeno 5 pacchetti in archivio \*\*\**

**SELECT Denominazione, COUNT (\*)  
FROM Nazioni, Pacchetti  
WHERE Nazioni.CodiceNaz = Pacchetti.CodiceNaz  
GROUP BY Denominazione;**

*\*\*\*viene restituito l'elenco di tutte le nazioni o localita' con il numero di pacchetti che si riferiscono ad esse. \*\*\**