



Programma 30 set 2015

- Installazione Macchina Virtuale
- Introduzione alla BI nelle Aziende
- Introduzione SQL



### Macchina Virtuale

- È un'emulazione di un computer su un altro computer
- Stesso punto di partenza per tutti
- SW necessari già installati
  - Win7
  - SQL Server Express 2012



### Macchina Virtuale

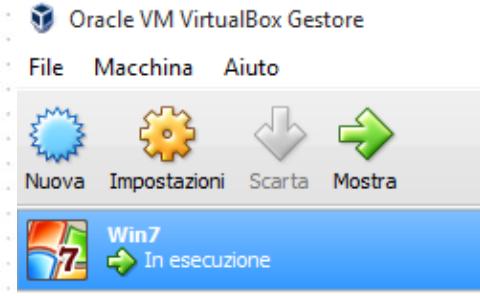
- Copiare il file Win7SIA.ova in locale sul proprio pc  
Richiede tempo → distribuire velocemente il file

#### Applicazione virtuale da importare

VirtualBox supporta attualmente l'importazione di applicazioni salvate in Open Virtualization Format (OVF). Per continuare, seleziona in basso il file da importare.



- Aprire l'applicazione Virtual Box
- Menu File → Importa Applicazione Virtuale
- Selezionare il file copiato
- Lanciare l'importazione
- Far partire la macchina virtuale



1



### Iscrizione alla mailing list, ai gruppi e comunicazioni con i docenti

- L'iscrizione alla mailing list avviene tramite il link <http://goo.gl/forms/KkLPUCVfnS> (è sul desktop della macchina virtuale)
  - La mail utilizzata è deve essere quella dell'università (se ne siete in possesso)
  - Possono essere inviate richieste ai docenti su:
    - [domenico.beneventano@unimore.it](mailto:domenico.beneventano@unimore.it)
    - [rp@waybi.it](mailto:rp@waybi.it)
    - [reggiosia@gmail.com](mailto:reggiosia@gmail.com)
  - L'oggetto della mail deve essere nel seguente formato:  
[SIARE15] [NOME GRUPPO] Oggetto
  - L'iscrizione ai gruppi avviene tramite il link  
<http://goo.gl/forms/Vo7Tjt91QI>



## Introduzione alla BI in Azienda



Come accedere su PC università

- Logout utente
- Login con PC[nr pc]\utentesql
- PW: sql



- Il linguaggio SQL è lo standard di fatto per interrogare i database
- Il nome deriva da Structured Query Language
- Nasce «Query» per le interrogazioni
- È alla base di praticamente tutte le estrazioni di dati
- Sintassi molto simile all'inglese
- Negli ultimi anni si stanno diffondendo linguaggi NoSQL



Il linguaggio SQL: Struttura

Si divide in tre sottoinsiemi:

- **Data Definition Language (DDL)** -- permette di creare e cancellare database o di modificarne la struttura  
Es: CREATE, DROP, ALTER
- **Data Manipulation Language (DML)** -- permette di inserire, cancellare e modificare i dati  
Es: SELECT, INSERT, UPDATE, DELETE
- **Data Control Language (DCL)** -- permette di gestire gli utenti e i permessi  
Es: GRANT, REVOKE



## DDL: Data Definition Language

Le istruzioni principali sono:

- CREATE: per creare nuovi database, tavole, indici, o viste.
- DROP: per cancellare esistenti database, tavole, indici, o viste.
- ALTER: per modificare un oggetto esistente nel database.
- TRUNCATE: per svuotare irreversibilmente una tabella.



## CREATE TABLE

Consente di creare una nuova tabella specificando uno o più campi accompagnati dal rispettivo tipo di dato.

CREATE TABLE <tabella> (

```
<campo> <tipo> [ NULL | NOT NULL ] [ <chiave> ] [ <vincolo> ],   <campo>
<tipo> [ NULL | NOT NULL ] [ <chiave> ] [ <vincolo> ],   ... )
```

- Tabella: nome della tabella
- Campo: nome di una colonna della tabella
- Tipo: specifica il tipo di dati della colonna (campo)
- NULL | NOT NULL: opzionale, indica se il campo ammette valori nulli o meno
- Chiave: opzionale, indica che il campo è una chiave (primary key o foreign key) della tabella
- Vincolo: vincoli aggiuntivi (e opzionali) ai campi



## Tipi di Dati

- **Char (n)**: dati carattere. Ha una **lunghezza fissa** (n) e può contenere fino ad 8000 caratteri ANSI
- **Varchar (n)**: dati carattere. Come il tipo Char può contenere fino a 8000 caratteri ANSI, ma ha una lunghezza variabile
- **Datetime**: Dati ora e data. Ammette valori compresi tra il 1 gennaio 1753 al 31 dicembre 9999 (precisione al trecentesimo di secondo)
- **Int**: Ammette i valori da -2147483648 a 2147483647
- **Numeric (p,s)**: Possono avere valori compresi tra  $10^{-38} + 1$  e  $-10^{38} - 1$ .
  - p - è la precisione, che rappresenta il numero massimo di cifre decimali che possono essere memorizzate (da entrambe le parti della virgola). Il massimo della precisione è 28 cifre.
  - s - è la scala, che rappresenta il numero di massimo di cifre decimali dopo la virgola e deve essere minore od uguale alla precisione.



## Definizione di Chiave [wiki]

### Primary key

La chiave primaria è un insieme di [attributi](#) che permette di individuare univocamente un [record](#) o [tupla](#) o [ennupla](#) in una tabella o [relazione](#).

L'esistenza della chiave primaria, che al più è formata da tutti gli attributi o campi dato della relazione, è garantita dall'unicità delle n-uple (o tuple) individuate da una [relazione](#) tra insiemi.

La chiave primaria è costituita da uno o più colonne della tabella (in questo caso viene definita chiave composta). La chiave primaria è unica per ogni tabella.

### Foreign key

Nel contesto dei [database relazionali](#), una **chiave esterna** (ingl. **foreign key**) è un [vincolo di integrità referenziale](#) tra due o più tabelle. Essa identifica una o più colonne di una tabella (referenziante) che referenzia una o più colonne di un'altra tabella (referenziata).

I valori di un record delle colonne referenzianti devono essere presenti in un unico record della tabella referenziata. Ciò implica che un record nella tabella referenziante non può contenere valori che non esistono nella tabella referenziata (eccetto nel caso particolare di valori [NULL](#)).

Più record della tabella referenziante possono puntare allo stesso record della tabella referenziata.

Nella maggior parte dei casi, ciò corrisponde alla relazione "uno a molti" tra una tabella padre ed una tabella figlio.



Esempio creazione tabella:

```
CREATE TABLE T_SEDI_REGIONALI      (
    CODICE_SEDE_REGIONALE INT NOT NULL PRIMARY KEY,
    SIGLA_REGIONE varchar(3) NOT NULL,
    DENOMINAZIONE varchar(60),
)
```

T_SEDI_REGIONALI		
CODICE SEDE REGIONALE	int	<pk>
SIGLA_REGIONE	varchar(3)	
DENOMINAZIONE	varchar(60)	



## DROP TABLE:

Comando utilizzato per eliminare una tabella e tutti i dati in essa contenuti

**DROP TABLE <nome tabella>**

Con l'eliminazione di una tabella, le regole o i valori predefiniti della tabella vengono disassociati e i vincoli associati alla tabella vengono eliminati automaticamente.



### Data Manipulation Language

Data Manipulation Language (DML) è un linguaggio di programmazione che consente di leggere, inserire, modificare o eliminare i dati in un database.

Le istruzioni principali del DML di SQL sono

- SELECT: esegue interrogazioni (query) sul DB
- INSERT: inserisce nuove tuple nel DB
- DELETE: cancella tuple dal DB
- UPDATE: modifica tuple del DB

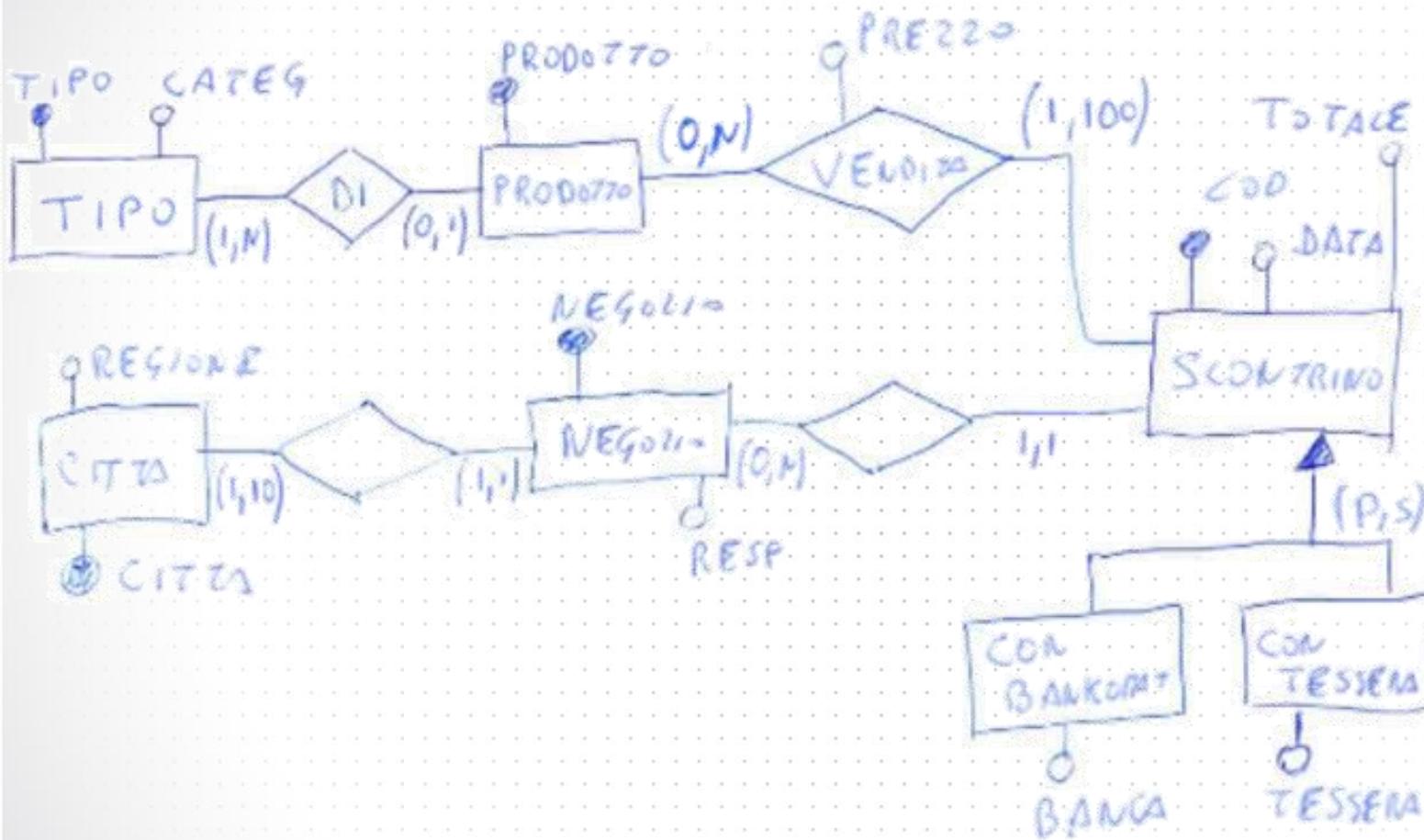


## Select

```
SELECT select_list
[ INTO new_table ]
FROM table_source
[ WHERE search_condition ]
[ GROUP BY group_by_expression ]
[ HAVING search_condition ]
[ ORDER BY order_expression [ ASC | DESC ] ]
```



## Esercizio Vendite





## Esercizio Vendite

### Schema Relazionale

TCITTA (CITTA, REGIONE)

TNEGOZIO (NEGOZIO, RESPONS, CITTA)

    FK CITTA REFERENCES TCITTA NOT NULL

TSCONTRINO (COD, DATA, TOTALE, NEGOZIO)

    FK NEGOZIO REFERENCES TNEGOZIO NOT NULL

TCONBANKOMAT (COD, BANCA)

    FK COD REFERENCES TSCONTRINO

TCONTESSERA (COD, TESSERA)

    FK COD REFERENCES TSCONTRINO

TTIPO (TIPO, CATEG)

TPRODOTTO (PRODOTTO, TIPO)

    FK TIPO REFERENCES TTIPO

VENDITA (COD, PRODOTTO, PREZZO)

    FK COD REFERENCES TSCONTRINO

    FK PRODOTTO REFERENCES TPRODOTTO

### Versione Semplificata

TCITTA (CITTA, REGIONE)

TNEGOZIO (NEGOZIO, RESPONS, CITTA:TCITTA)

TSCONTRINO (COD, DATA, TOTALE, NEGOZIO:TNEGOZIO)

TCONBANKOMAT (COD:TSCONTRINO, BANCA)

TCONTESSERA (COD:TSCONTRINO, TESSERA)

TTIPO (TIPO, CATEG)

TPRODOTTO (PRODOTTO, TIPO:TTIPO)

VENDITA (COD:TSCONTRINO, PRODOTTO:TPRODOTTO, PREZZO)

*La versione semplificata è quella in cui la Foreign Key viene indicata direttamente accanto al nome dell'attributo*



## Esercizio Vendite

```
CREATE TABLE TCITTA(
    CITTA CHAR(12),
    REGIONE CHAR(12),
    PRIMARY KEY(CITTA))

CREATE TABLE TNEGOZIO(
    NEGOZIO CHAR(12),
    RESPONS CHAR(12),
    CITTA CHAR(12) NOT NULL, -- LA FOREIGN KEY CITTA NON PUO' ESSERE NULL
    PRIMARY KEY(NEGOZIO),
    FOREIGN KEY(CITTA) REFERENCES TCITTA)

CREATE TABLE TS CONTRINO(
    COD INT,
    DATA DATETIME,
    TOTALE FLOAT,
    NEGOZIO CHAR(12) NOT NULL, -- LA FOREIGN KEY NEGOZIO NON PUO' ESSERE NULL
    PRIMARY KEY(COD),
    FOREIGN KEY(NEGOZIO) REFERENCES TNEGOZIO )

CREATE TABLE TCONBANKOMAT(
    COD INT, BANCA CHAR(12),
    PRIMARY KEY(COD),
    FOREIGN KEY(COD) REFERENCES TS CONTRINO)

CREATE TABLE TCONTESSERA(
    COD INT, TESSERA CHAR(12),
    PRIMARY KEY(COD),
    FOREIGN KEY(COD) REFERENCES TS CONTRINO)
```



## Esercizio Vendite

```
CREATE TABLE TPRODOTTO(
    PRODOTTO CHAR(12),
    TIPO CHAR(12), -- LA FOREIGN KEY TIPO PUO' ESSERE NULL, CI SONO PRODOTTI SENZA TIPO
    PRIMARY KEY (PRODOTTO),
    FOREIGN KEY (TIPO) REFERENCES TTIPO )
```

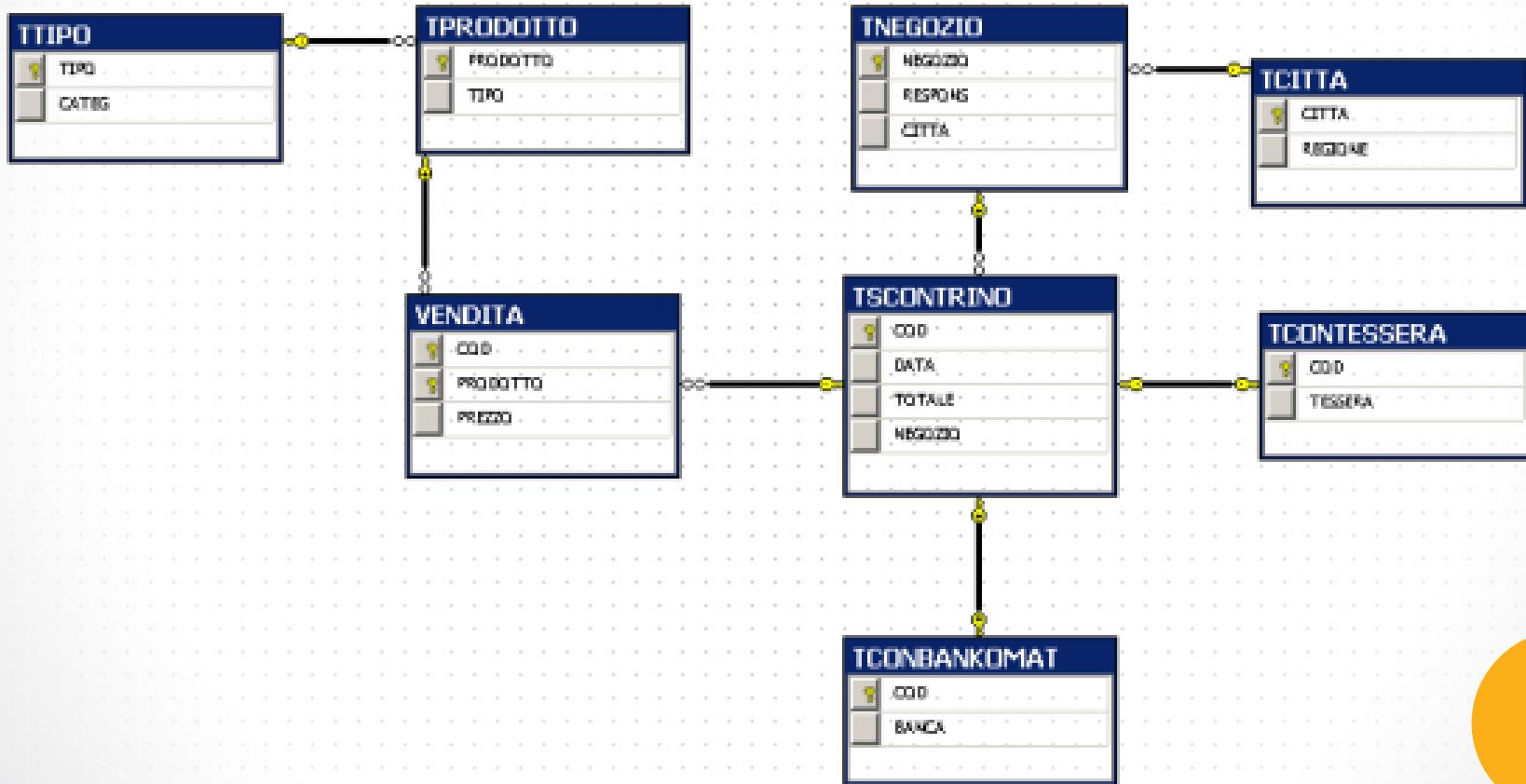
```
CREATE TABLE VENDITA(
    COD INT,
    PRODOTTO CHAR(12),
    PREZZO INT,
    PRIMARY KEY (COD, PRODOTTO),
    FOREIGN KEY (COD) REFERENCES TSCONTRINO,
    FOREIGN KEY (PRODOTTO ) REFERENCES TPRODOTTO )
```



## Esercizio Vendite

Diagramma relazionale:

È possibile definire le tabelle senza e chiavi esterne e indicarle successivamente direttamente nello schema





```
use SIATEST

CREATE TABLE TCITTA(
CITTA CHAR(12),
REGIONE CHAR(12),
PRIMARY KEY(CITTA))

CREATE TABLE TNEGOZIO(
NEGOZIO CHAR(12),
RESPONS CHAR(12),
CITTA CHAR(12) NOT NULL, -- LA FOREIGN KEY CITTA NON PUO' ESSERE NULL
PRIMARY KEY(NEGOZIO),
FOREIGN KEY(CITTA) REFERENCES TCITTA)

CREATE TABLE TSCONTRINO(
COD INT,
DATA DATETIME,
TOTALE FLOAT,
NEGOZIO CHAR(12) NOT NULL, -- LA FOREIGN KEY NEGOZIO NON PUO' ESSERE NULL
PRIMARY KEY(COD),
FOREIGN KEY(NEGOZIO) REFERENCES TNEGOZIO )

CREATE TABLE TCONBANKOMAT(
COD INT, BANCA CHAR(12),
PRIMARY KEY(COD),
FOREIGN KEY(COD) REFERENCES TSCONTRINO)

CREATE TABLE TCONTESSERA(
COD INT, TESSERA CHAR(12),
PRIMARY KEY(COD),
FOREIGN KEY(COD) REFERENCES TSCONTRINO)

CREATE TABLE TTIPPO(
TIPO CHAR(12), CATEG CHAR(12),
PRIMARY KEY(TIPO))

CREATE TABLE TPRODOTTO(
PRODOTTO CHAR(12),
TIPO CHAR(12), -- LA FOREIGN KEY TIPO PUO' ESSERE NULL, CI SONO PRODOTTI SENZA TIPO
PRIMARY KEY(PRODOTTO),
FOREIGN KEY(TIPO) REFERENCES TTIPPO )

CREATE TABLE VENDITA(
COD INT,
PRODOTTO CHAR(12),
PREZZO INT,
PRIMARY KEY(COD,PRODOTTO),
FOREIGN KEY(COD) REFERENCES TSCONTRINO,
FOREIGN KEY(PRODOTTO ) REFERENCES TPRODOTTO )
```

## Esercizio Vendite Testo script