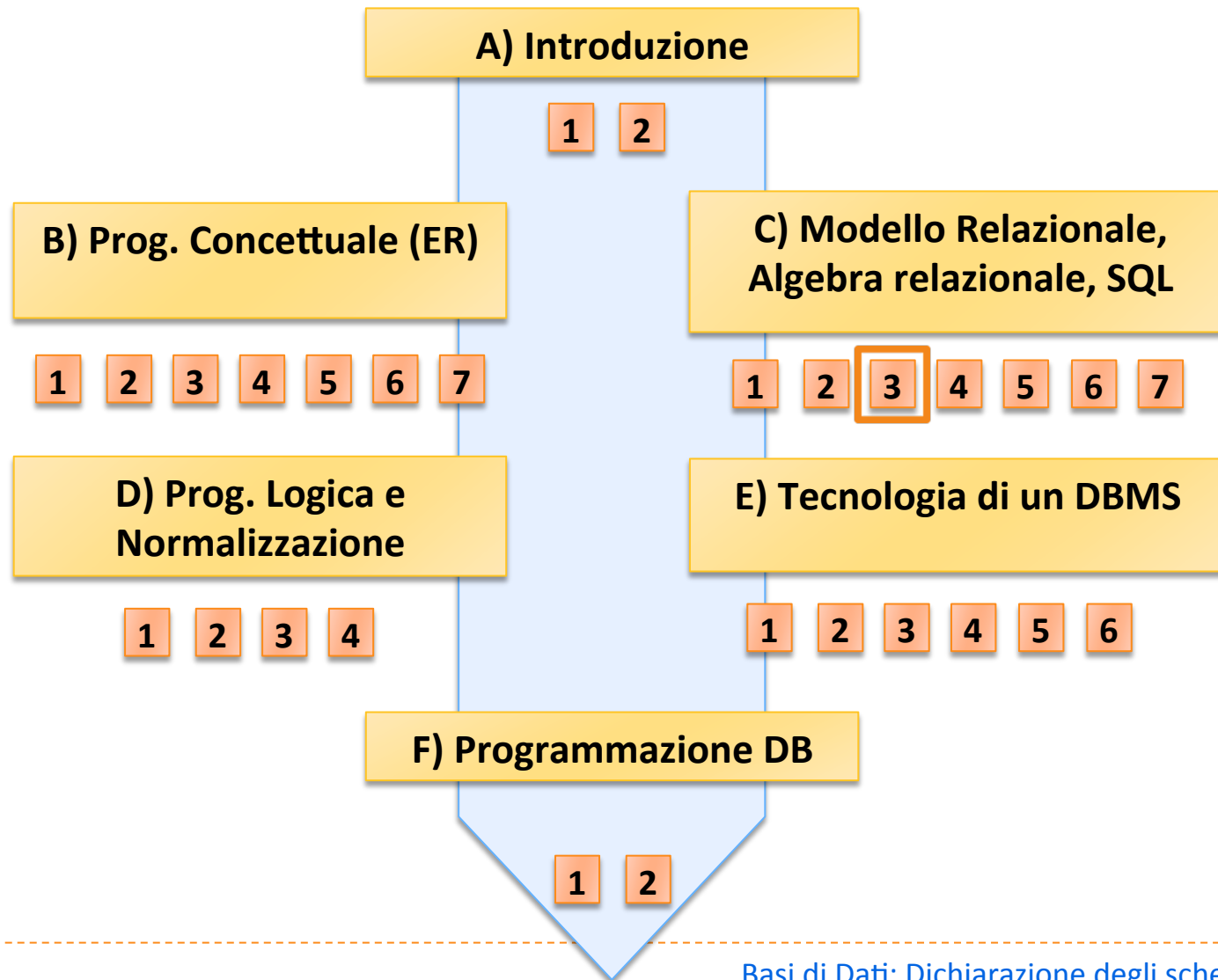


Basi di Dati

Dichiarazione degli schemi in SQL

Basi di Dati – Dove ci troviamo?



Storia di SQL

- ▶ Definito nell'ambito del progetto
 - ▶ **SYSTEM R** (IBM S. JOSE) nel 1976
- ▶ Nome originario: **SEQUEL**
- ▶ Adottato progressivamente da tutti i sistemi commerciali
- ▶ Standardizzato da **ANSI** e **ISO**

Composizione di SQL

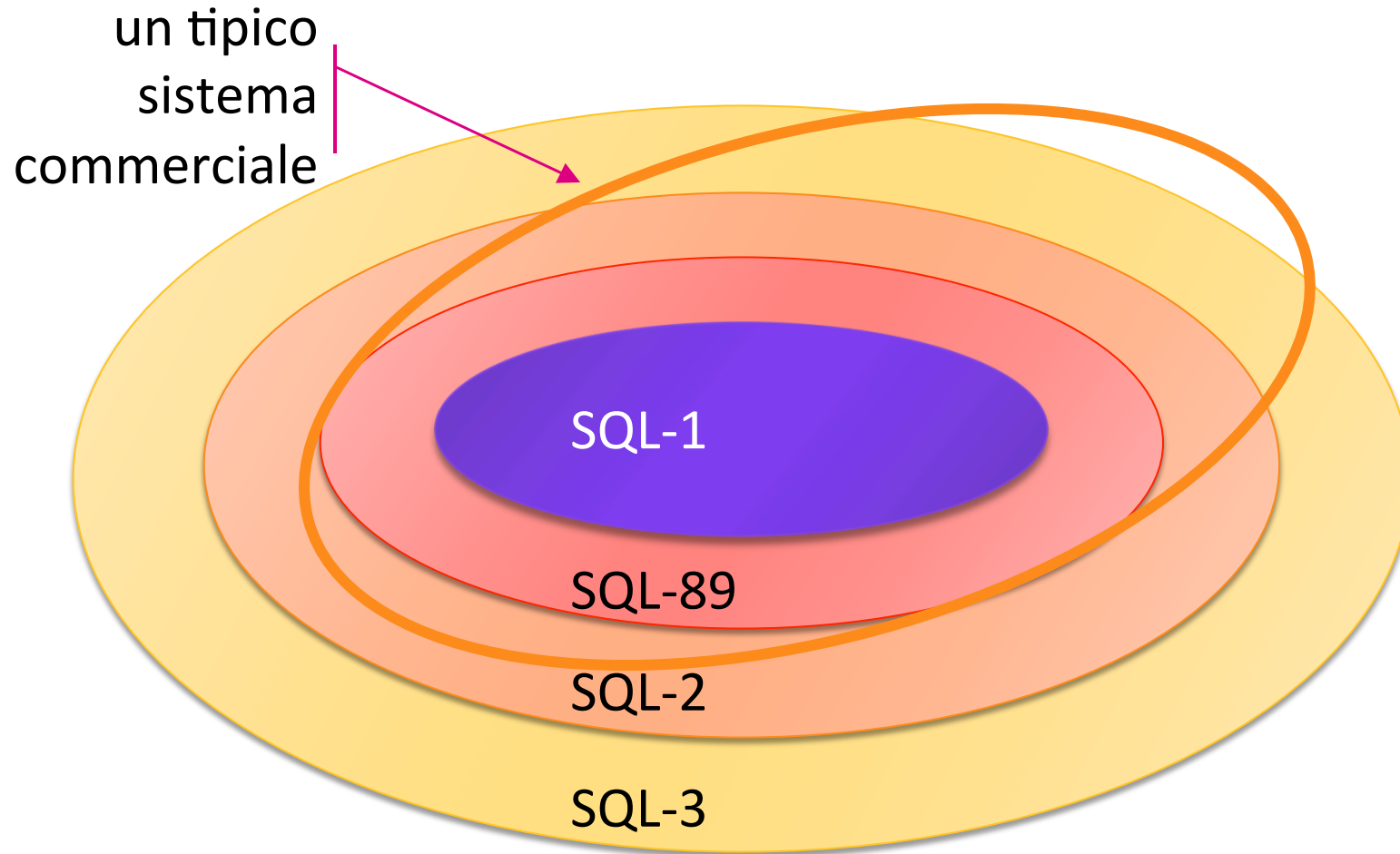
- ▶ Data Definition Language (DDL)
 - ▶ definizione di domini, tabelle, indici, viste, vincoli, procedure, trigger
- ▶ Data Manipulation Language (DML)
 - ▶ linguaggio di query, modifica, comandi transazionali
- ▶ Data Control Language (DCL)
 - ▶ linguaggio per la gestione degli utenti e dei privilegi di accesso

Standardizzazione di SQL

- ▶ Progressione dello standard per estensioni quasi-compatibili
 - ▶ prima versione nel 1986 (SQL-1)
 - ▶ modifica alla prima versione nel 1989 (SQL-89)
 - ▶ seconda versione nel 1992 (SQL-2, SQL-92)
 - ▶ terza versione nel 1999 (SQL-3, SQL-99)

- ▶ In SQL-2:
 - ▶ entry SQL (più o meno equivalente a SQL-89)
 - ▶ intermediate SQL
 - ▶ full SQL

Potere espressivo di standard e sistemi commerciali



Tipi di dati in SQL-2

stringhe

- ▶ CHAR (N)
- ▶ VARCHAR (N)

stringhe di bit (0/1)

- ▶ BIT (N)
- ▶ VARBIT (N)

numerici esatti

- ▶ NUMERIC (Prec, Scale)
(o anche DECIMAL)
- ▶ INTEGER
- ▶ SMALLINT

Precision e Scale

Esempio

23.5141

- ▶ Precision: 6
(numero di cifre significative in tutto il numero)
- ▶ Scale: 4
(numero di cifre decimali)

Tipi di dati in SQL-2 (cont.)

Numerici approssimati

- ▶ REAL

- ▶ DOUBLE PRECISION

Domini speciali

- ▶ DATE
 - ▶ YYYY-MM-DD

- ▶ TIME(N)
 - ▶ HH:MM:SS.NNNN

- ▶ TIMESTAMP
 - ▶ YYYY-MM-DD HH:MM:SS

- ▶ INTERVAL
 - ▶ INTERVAL YEAR(2) TO MONTH
 - ▶ (tra 0 anni – 0 mesi e 99 anni – 11 mesi)

Il valore "null"

- ▶ **null** è un valore polimorfo (che appartiene a tutti i domini) col significato di valore non noto
- ▶ il valore esiste in realtà ma è **ignoto** al database
 - ▶ Es: data di nascita
- ▶ il valore è **inapplicabile**
 - ▶ Es: numero patente per minorenni

Definizione dei domini applicativi

- ▶ Un dominio è un insieme di valori consentiti
- ▶ Un dominio può essere un tipo di dato base dell'SQL (**dominio elementare**) oppure può essere **definito dall'utente**:

CREATE DOMAIN <nome-dominio>

AS <tipo-di-dati>

[**DEFAULT**<valore-default>]

[<vincoli-di-dominio>]

Esempi

```
CREATE DOMAIN PrezzoQuotidiani  
AS DECIMAL  
DEFAULT 1,1  
NOT NULL
```

```
CREATE DOMAIN OreLezione  
AS SMALLINT  
DEFAULT 80
```

Definizione delle tabelle

- ▶ Una tabella è costituita da:
 - ▶ una lista di uno o più attributi (colonne)
 - ▶ un insieme di zero o più vincoli

```
CREATE TABLE <nome-tabella>  
(<nome-col><dominio> [<vincoli-col>] ,  
  ...  
<nome-col><dominio> [<vincoli-col>] ,  
  [<vincoli-tab>]  
)
```

Vincoli di colonna

- ▶ **NOT NULL**
 - ▶ l'attributo non può assumere il valore null
- ▶ **UNIQUE**
 - ▶ unicità dell'attributo
- ▶ **PRIMARY KEY**
 - ▶ l'attributo è la chiave primaria
- ▶ **CHECK**
 - ▶ esprime un generico vincolo sulla colonna tramite una espressione logico-relazionale
- ▶ **REFERENCES**
 - ▶ esprime il vincolo della Foreign Key

Vincoli di tabella

- ▶ UNIQUE (<lista-colonne>)
 - ▶ la combinazione dei valori delle colonne deve essere unica per tutte le tuple della tabella
- ▶ PRIMARY KEY (<lista-colonne>)
 - ▶ chiave primaria della tabella (implica NOT NULL)
- ▶ FOREIGN KEY (<lista-colonne>) REFERENCES <tab> [(<lista-colonne>)]
 - ▶ foreign key della tabella
- ▶ CHECK (<condizione>)
 - ▶ predicato che deve essere soddisfatto per tutte le tuple della tabella

Esempi

CREATE TABLE STUDENTE

```
(  MATR CHAR(6) PRIMARY KEY,  
  NOME VARCHAR (30) NOT NULL,  
  CITTÀ VARCHAR (20),  
  C-DIP CHAR (3) )
```

CREATE TABLE CORSO

```
(  COD-CORSO CHAR(6) PRIMARY KEY,  
  TITOLO VARCHAR(30) NOT NULL,  
  DOCENTE VARCHAR(20) )
```


Esempi (cont.)

CREATE TABLE ESAME

(**MATR** CHAR(6),

COD-CORSO CHAR(6),

DATA DATE NOT NULL,

VOTO SMALLINT NOT NULL,

 PRIMARY KEY(MATR,COD-CORSO),

 FOREIGN KEY (...) (v. slide successivi)

)

Chiavi alternative

- ▶ Come si esprime una chiave alternativa?
 - ▶ Specificando le clausole **NOT NULL** e **UNIQUE**

CREATE TABLE STUDENTE

```
(  MATR CHAR(6) PRIMARY KEY,  
  CF CHAR(16) NOT NULL UNIQUE,  
  NOME VARCHAR (30) NOT NULL,  
  CITTÀ VARCHAR (20),  
  C-DIP CHAR (3) )
```



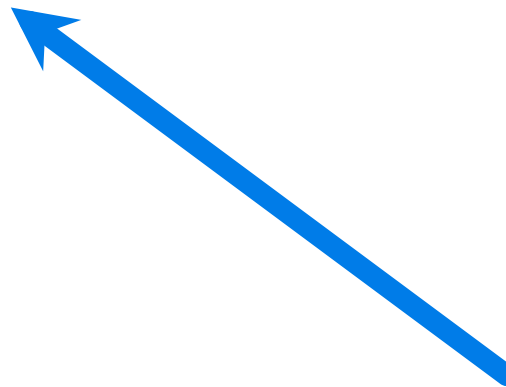
Integrità referenziale

- ▶ Esprime un legame gerarchico (padre-figlio) fra tabelle
- ▶ Alcuni attributi della tabella figlio sono definiti **FOREIGN KEY**
- ▶ I valori contenuti nella **FOREIGN KEY** devono essere sempre presenti nella tabella padre

Esempio: studente - esame

studente

MATR	
123	
415	
702	



esame

MATR	
123	
123	
702	

Il problema degli orfani

studente

MATR	
123	
415	
702	

ORFANI:

tuple che restano prive di padre a causa di cancellazioni e modifiche della tabella padre

esame

MATR	
123	
123	
702	

Gestione degli orfani

- ▶ Cosa succede degli esami se si cancellano gli studenti?
 - ▶ CASCADE (si cancellano anche gli esami)
 - ▶ SET NULL
 - ▶ SET DEFAULT
 - ▶ NO ACTION

- ▶ Cose succede degli esami se si modifica la matricola nella tabella STUDENTE?
 - ▶ CASCADE (si modificano anche gli esami)
 - ▶ SET NULL
 - ▶ SET DEFAULT
 - ▶ NO ACTION

Gestione degli orfani

▶ CASCADE

- ▶ Aggiornamenti su colonne riferite aggiornano tutte le colonne delle tuple con foreign key che si riferiscono ad esse
- ▶ Cancellazioni di tuple riferite cancellano tutte le tuple contenenti riferimenti ad esse

▶ SET NULL

- ▶ Aggiornamenti e cancellazioni su colonne riferite causano la modifica delle colonne di foreign key a NULL

Gestione degli orfani

- ▶ SET DEFAULT

- ▶ Aggiornamenti e cancellazioni su colonne riferite causano la modifica delle colonne di foreign key al valore di default

- ▶ NO ACTION

- ▶ Aggiornamenti e cancellazioni su colonne riferite sono proibiti se riferiti da almeno una tupla con foreign key

Definizione : nella tabella figlia

```
CREATE TABLE ESAME
(
    .....
    FOREIGN KEY MATR REFERENCES STUDENTI
        ON DELETE CASCADE ON UPDATE CASCADE )
```

- ▶ È lecito essere figli di più padri

```
CREATE TABLE ESAME
(
    ....
    PRIMARY KEY(MATR,COD-CORSO) ,
    FOREIGN KEY MATR REFERENCES STUDENTI
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY COD-CORSO REFERENCES CORSO
        ON DELETE NO ACTION ON UPDATE NO ACTION )
```

Una istanza scorretta

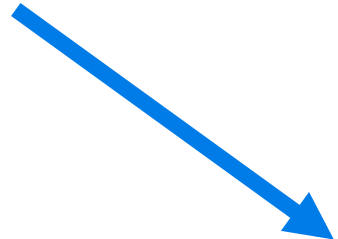
studente

MATR	NOME	CITTÀ	C-DIP
123			
415			
702			

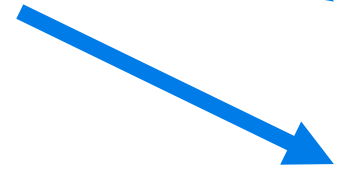
esame

MATR	COD-CORSO	DATA	VOTO
123	1	2014-09-07	30
123	2	2015-01-08	28
123	2	2014-08-01	28
702	2	2014-09-07	20
702	1	NULL	NULL
714	1	2014-09-07	28

viola la chiave



viola il NOT NULL



viola la integrità referenziale



Una istanza corretta

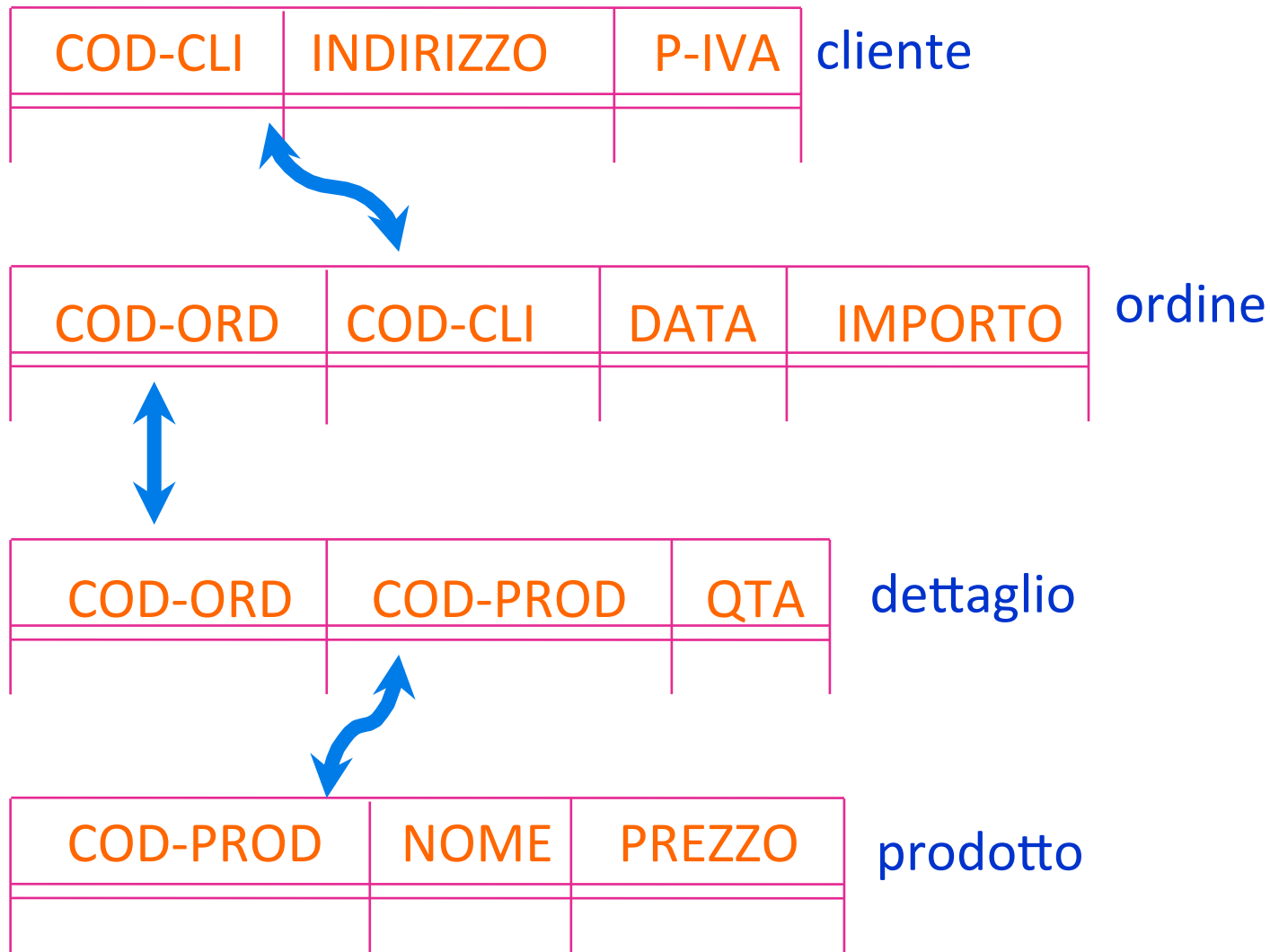
studente

MATR	NOME	CITTÀ	C-DIP
123			
415			
702			

esame

MATR	COD-CORSO	DATA	VOTO
123	1	2014-09-07	30
123	2	2015-01-08	28
702	2	2014-09-07	20

Esempio : gestione ordini



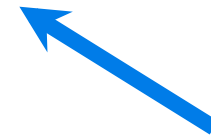
Definizione della tabella CLIENTE

cliente

COD-CLI	INDIRIZZO	P-IVA

CREATE TABLE **CLIENTE**

(
 COD-CLI CHAR(6) PRIMARY KEY,
 INDIRIZZO CHAR(50),
 P-IVA CHAR(12) NOT NULL UNIQUE
)



Chiave alternativa

Definizione della tabella ORDINE

ordine

COD-ORD	COD-CLI	DATA	IMPORTO

```
CREATE TABLE ORDINE
(
  COD-ORD CHAR(6) PRIMARY KEY,
  COD-CLI CHAR(6) NOT NULL DEFAULT='999999',
  DATA DATE,
  IMPORTO INTEGER,
  FOREIGN KEY COD-CLI REFERENCES CLIENTE
    ON DELETE SET DEFAULT
    ON UPDATE SET DEFAULT
)
```

Definizione della tabella DETTAGLIO

dettaglio

COD-ORD	COD-PROD	QTA

```
CREATE TABLE DETTAGLIO
(
  COD-ORD CHAR(6),
  COD-PROD CHAR(6),
  QTA SMALLINT,
  PRIMARY KEY (COD-ORD,COD-PROD),
  FOREIGN KEY COD-ORD REFERENCES ORDINE
    ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY COD-PROD REFERENCES PRODOTTO
    ON DELETE NO ACTION ON UPDATE NO ACTION
)
```

Definizione della tabella PRODOTTO

prodotto

COD-PROD	NOME	PREZZO

```
CREATE TABLE PRODOTTO  
(  
  COD-PROD CHAR(6) PRIMARY KEY,  
  NOME CHAR(20),  
  PREZZO SMALLINT  
)
```


Esercizio: gestione personale

- ▶ esprimere in SQL la dichiarazione dello schema

MATR	NOME	DATA-ASS	SALARIO	MATR-MGR
1	Piero	2012-01-01	1500 €	2
2	Giorgio	2014-01-01	2000 €	null
3	Giovanni	2013-07-01	1000 €	2

impiegato

assegnamento

MATR	NUM-PROG	PERC
1	3	50
1	4	50
2	3	100
3	4	100

progetto

NUM-PROG	TITOLO	TIPO
3	Idea	Esprit
4	Wide	Esprit