

Università degli Studi di Trieste

Corso di Laurea Magistrale in
INGEGNERIA CLINICA

ESERCITAZIONE: IL LINGUAGGIO SQL

Corso di Informatica Medica
Docente Sara Renata Francesca MARCEGLIA



Dipartimento di Ingegneria e Architettura



UNIVERSITÀ
DEGLI STUDI DI TRIESTE

SQL



- SQL (**S**tructured **Q**uery **L**anguage): linguaggio di interrogazione per basi di dati relazionali
- SQL è un **ISO/ANSI standard**: indipendente dal DBMS su cui è implementato;
- **SQL è un LINGUAGGIO DICHIARATIVO**: specifica le **CARATTERISTICHE DEL RISULTATO** e NON la **PROCEDURA** per ottenerlo;
- SQL utilizza i termini **Tabella, Riga, Colonna** che corrispondono a Relazione, Tupla, Attributo nel modello relazionale.
- SQL definisce le operazioni di
 - Definizioni di dati (schema e istanze)
 - Istruzioni di aggiornamento (schema e istanze)
 - Interrogazioni



GLOSSARIO

- **dominio**: insieme dei valori ammissibili che una colonna (un attributo) può assumere;
- **tabella**: insieme ordinato di colonne ed eventuali vincoli relativi ai valori degli attributi stessi;
- **indice**: struttura dati ausiliaria, associata ad una tabella, che permette di rendere più efficiente l'esecuzione di interrogazioni sulla tabella mediante l'introduzione di un **ordinamento** sui valori di uno o più attributi;



GLOSSARIO

- **asserzione**: rappresenta dei vincoli non legati ad attributi o tabelle, ma direttamente connessi allo schema della base di dati;
- **vista**: è una tabella “virtuale”, il cui contenuto è legato al contenuto di altre tabelle della base di dati;
- **privilegio**: definizione delle modalità di accesso ai dati da parte di un certo utente.

CREAZIONE DEL DATABASE



- Istruzione per la creazione di una nuova base di dati

CREATE DATABASE <nomeDataBase>;

- Istruzione per la cancellazione di una base di dati

DROP DATABASE <nomeDataBase>;



CREAZIONE TABELLA

- Istruzione di creazione

```
CREATE TABLE <nomeTabella> (  
    <nomeAtt1> <TIPOATTRIBUTO> [DEFAULT] [VINCOLI],  
    <nomeAtt2> <TIPOATTRIBUTO> [DEFAULT] [VINCOLI],  
    ...  
    <nomeAttN> <TIPOATTRIBUTO> [DEFAULT] [VINCOLI],  
  
    [ALTRI VINCOLI – es. Integrità referenziale]  
);
```



TIPI DI DATO PRINCIPALI

- **INTERO: INT**
- **DECIMALE: DECIMAL (M,N)**
 - M=numero totale di cifre
 - N= numero di cifre dopo la virgola
 - **FLOAT** – approssimato precisione 16 cifre
- **STRINGA:**
 - **CHARACTER (N)** – stringa di lunghezza fissa N
 - **VARCHAR (N)** – stringa di lunghezza variabile, massimo N caratteri
- **BOOLEANO: BOOLEAN**
- **BINARIO**
 - **BINARY (N)** – binario di lunghezza fissa N
 - **VARBINARY (N)** – binario di lunghezza variabile, massimo N
- **DATA**
 - **DATE - 'YYYY-MM-DD'**
 - **DATETIME 'YYYY-MM-DD HH:MM:SS'**
 - **TIMESTAMP**

CREAZIONE TABELLA PAZIENTE



PatientID	Nome	Cognome	Sesso	Data Nascita	CF
1	Anna	Rossi	F	11/3/62	RSSNNA62C51L872P
2	Roberto	Marzio	M	4/25/71	MRZRRT71D04F251R
3	Giampiero	Di Nicola	M	7/3/37	DNCGPR37L07H821Q
4	Luciana	Nunziatella	F	5/11/55	NNZLCN55R45F205N
5	Arianna	Lucchini	F	2/2/82	LCCRNN82B42N127C

```
CREATE TABLE paziente (  
patientID INT PRIMARY KEY AUTO_INCREMENT,  
Nome VARCHAR (20),  
Cognome VARCHAR (20),  
DataNascita DATE DEFAULT NULL  
);
```




MODIFICA TABELLA

- Cancellazione tabella

DROP TABLE <NomeTabella>

- Visualizzazione schema

DESCRIBE <NomeTabella>

- Modifica tabella: aggiungere una Colonna

ALTER TABLE <NomeTabella> **ADD** <nomeAtt> <Tipo>

- Modifica tabella:eliminare una Colonna

ALTER TABLE <NomeColonna> **DROP COLUMN** <nomeAtt>;

ALTER TABLE paziente *ADD* cf *VARCHAR*(16) *UNIQUE*;



INSERIMENTO DATI

- Istruzione base

INSERT INTO <nomeTabella> **VALUES** (Val1, Val2, ..., ValN)

N=numero attributi

- Inserimento selettivo

INSERT INTO <nomeTabella> (*Att1, Att2,...*) **VALUES** (*Val1,Val2,...*)

Si inseriscono solo i valori dichiarati nella lista (*Att1,Att2, ...*) nell'ordine dichiarati

```
INSERT INTO paziente VALUES (1,'Anna','Rossi', 'F','1962-3-11','RSSNNA62C51L872P');
```

```
INSERT INTO paziente (Nome, Cognome, Sesso, DataNascita,cf)  
VALUES ('Roberto','Marzio','M','1971-04-25','MRZRRT71D04F251R');
```



VINCOLI

- **PRIMARY KEY** – l'attributo è la chiave primaria della tabella
- **NOT NULL** – l'attributo deve essere sempre dichiarato (è sottinteso in PRIMARY KEY)
- **UNIQUE** – l'attributo deve essere univoco
- **AUTO_INCREMENT** – tipico dell'indice che è Primary key della tabella (fa aumentare l'indice automaticamente senza doverlo inserire)
- **DEFAULT** - valore di default che l'attributo deve assumere



MODIFICA VALORI

- AGGIORNAMENTO VALORE (Istruzione base)

UPDATE <NomeTabella>

SET <NomeAtt1> = VALORE1 {, <NomeAtt2> = VALORE2,... <NomeAttN> = VALOREN }

{**WHERE** [condizione] };

In assenza della clausola WHERE la modifica ricade su tutte le righe

Possono essere modificate più colonne contemporaneamente

- CANCELLAZIONE RIGA

DELETE FROM <NomeTabella>

{**WHERE** [condizione]};



INTERROGAZIONE: SINTASSI BASE

Modello di Interrogazione generica

SELECT <lista di attributi>

FROM <lista delle tabelle>

WHERE <condizione>

```
SELECT Nome,Cognome  
FROM paziente  
WHERE Nome='Anna';
```

DAL LINGUAGGIO NATURALE ALLA QUERY



Traduzione di una query di select in linguaggio naturale
(traduzione linguaggio dichiarativo in procedurale):

1. Tra le righe ottenute dal prodotto cartesiano delle tabelle elencate nella clausola **from**,
2. vengono considerate quelle righe che soddisfano la condizione espressa nella clausola **where**;
3. su tali righe vengono valutate le espressioni sulle colonne indicate nella clausole **select**.

INTERROGAZIONE: OPZIONI DI VISUALIZZAZIONE



- ORDER BY – presenta i risultati secondo un ordinamento

ORDER BY <NomeAtt> <TIPOLOGIA>

TIPOLOGIA = ASC (default), DESC

- LIMIT – presenta un numero limitato di righe

LIMIT N

N= numero massimo di righe

- AS – modifica il nome della Colonna (in visualizzazione)

SELECT <NomeAtt> **AS** <NuovoNome>

VISUALIZZAZIONE TABELLA



```
SELECT *  
FROM paziente;
```

Success 2:22 PM
5 rows 0.588 seconds

Explore SQL **Data** Chart Export ▾

patientID	Nome	Cognome	Sesso	DataNascita	cf
1	Anna	Rossi	F	1962-11-03	RSSNNA62C51L872P
2	Roberto	Marzio	M	1971-04-25	MRZRRT71D04F251R
3	Giampiero	Di Nicola	M	1937-07-03	DNCGPR37L07H821Q
4	Luciana	Nunziatella	F	1955-05-11	NNZLCN55R45F205N
5	Arianna	Lucchini	F	1982-02-02	LCCRNN82B42N127C



Queries 1.1, 1.2 e 1.3

- 1.1 Conoscere cognome e nome dei pazienti maschi considerati nella base di dati;
- 1.2 Conoscere tutti i dati relativi ai pazienti il cui cognome è “Nunziatella”;
- 1.3 Conoscere cognome, nome, anno di nascita dei pazienti considerati.



Queries 1.1, 1.2 e 1.3

1.1 Conoscere cognome e nome dei pazienti maschi considerati nella base di dati;

$1.1 \leftarrow \pi_{(\text{Cognome}, \text{Nome})} \sigma_{(\text{Sesso}='M')} \text{paziente}$

1.2 Conoscere tutti i dati relativi ai pazienti il cui cognome è “Nunziatella”;

$1.2 \leftarrow \sigma_{(\text{Cognome}='Nunziatella')} \text{Paziente}$

1.3 Conoscere cognome, nome, anno di nascita dei pazienti considerati.

operatore $\text{YEAR}(\text{'YYYY-MM-DD'}) \rightarrow \text{YYYY}$

$1.3 \leftarrow \pi_{(\text{Cognome}, \text{Nome}, \text{year}(\text{DataNascita}))} \text{Paziente}$



Query 1.1 : implementazione

```
SELECT Nome,Cognome  
FROM paziente  
WHERE Sesso='M';
```

Success
2 rows 0.572 s

Explore SQL Data Chart Export ▾

Nome	Cognome
Roberto	Marzio
Giampiero	Di Nicola



Query 1.2 : implementazione

```
SELECT *  
FROM paziente  
WHERE Cognome='Nunziatella';
```

Success 2
1 rows 0.569 s

Explore SQL **Data** Chart Export ▾ 🔗

patientID	Nome	Cognome	Sesso	DataNascita	cf
4	Luciana	Nunziatella	F	1955-05-11	NNZLCN55R45F205N



Query 1.3 : implementazione

```
SELECT Cognome, Nome, YEAR(DataNascita) AS Anno_Di_Nascita  
FROM paziente;
```

Success 2:24 PM
5 rows 0.57 seconds

Explore SQL **Data** Chart Export ▾ 🔗

Cognome	Nome	Anno_Di_Nascita
Rossi	Anna	1962
Marzio	Roberto	1971
Di Nicola	Giampiero	1937
Nunziatella	Luciana	1955
Lucchini	Arianna	1982



Query 1.3bis

```
SELECT Cognome, Nome, YEAR(DataNascita) AS Anno_Di_Nascita  
FROM paziente  
ORDER BY(Cognome) DESC;
```

Success

5 rows

Cognome	Nome	Anno_Di_Nascita
Rossi	Anna	1962
Nunziatella	Luciana	1955
Marzio	Roberto	1971
Lucchini	Arianna	1982
Di Nicola	Giampiero	1937

RELAZIONI TRA TABELLE: VINCOLI DI INTEGRITÀ REFERENZIALE



patientID	Nome	Cognome	Sesso	DataNascita	cf	MedicoID	
1	Anna	Rossi	F	1962-03-11	RSSNNA62C51L872P	1	PAZIENTE
2	Roberto	Marzio	M	1971-04-25	MRZRRT71D04F251R	NULL	
3	Giampiero	Di Nicola	M	1937-07-03	DNCGPR37L07H821Q	NULL	
4	Luciana	Nunziatella	F	1955-05-11	NNZLCN55R45F205N	1	
5	Arianna	Lucchini	F	1982-02-02	LCCRNN82B42N127C	1	
6	MariaLuisa	Gichetti	F	1953-08-25	GCTMLS53H65H832F	1	

therapyID	patientID	Nome	CasaProduttrice	costo	PrincipioAttivo	DataInizio	DataFine	
1	3	Cumadin	Bayer	45.00	warfarin	2003-07-15	NULL	TERAPIA
2	1	Corvel	Dompe	3.00	carvedilolo	2015-01-17	2019-03-15	
3	2	Enapren	Zambon	5.00	enelapril	1999-10-11	2009-11-03	
4	5	Corvel	Dompe	3.00	carvedilolo	2007-03-04	2017-09-16	
5	4	Cumadin	Bayer	45.00	warfarin	2000-04-27	2010-10-23	
6	2	Cumadin	Bayer	45.00	warfarin	2009-11-04	2019-09-21	
7	5	Nebilox	Lobivon	7.00	Nebivololo	2016-12-01	2019-09-21	

employeeID	Nome	Cognome	Sesso	matricola	qualifica	
1	Bruno	Mascheroni	M	12345	dirigente medico	DIPENDENTI
2	Anna	Trilli	F	61549	dirigente medico	
3	Vincenzo	Cinti	M	95346	infermiere	
4	MariaChiara	Bandai	F	87309	tecnico radiologia	
5	Andrea	Giannoni	F	25685	infermiere	



CREAZIONE TABELLA TERAPIA

```
CREATE TABLE terapia (  
therapyID INT PRIMARY KEY AUTO_INCREMENT,  
patientID INT,  
Nome VARCHAR(30),  
CasaProduttrice VARCHAR(30),  
costo DECIMAL(5,2),  
PrincipioAttivo VARCHAR(49),  
DataInizio DATE,  
DataFine DATE,  
FOREIGN KEY(patientID) REFERENCES paziente(patientID) ON DELETE  
CASCADE  
  
);
```

```
INSERT INTO terapia(patientID, Nome, CasaProduttrice, costo, PrincipioAttivo, DataInizio, DataFine)  
VALUES (3, 'Cumadin', 'Bayer', 45.00, 'warfarin', '2003-07-15', NULL);
```


CREAZIONE TABELLA DIPENDENTI E COLLEGAMENTO A PAZIENTE



```
CREATE TABLE dipendenti(  
employeeID INT PRIMARY KEY AUTO_INCREMENT,  
Nome VARCHAR(20),  
Cognome VARCHAR(20),  
Sesso VARCHAR(1),  
matricola INT,  
qualifica VARCHAR(20)  
);
```

```
INSERT INTO dipendenti (Nome,Cognome,Sesso,matricola,qualifica) VALUES ('Bruno','Mascheroni','M',12345,'dirigente medico');  
INSERT INTO dipendenti (Nome,Cognome,Sesso,matricola,qualifica) VALUES ('Anna','Trilli','F',61549,'dirigente medico');  
INSERT INTO dipendenti (Nome,Cognome,Sesso,matricola,qualifica) VALUES ('Vincenzo','Cinti','M',95346,'infermiere');  
INSERT INTO dipendenti (Nome,Cognome,Sesso,matricola,qualifica) VALUES ('MariaChiara','Bandai','F',87309,'tecnico radiologia');  
INSERT INTO dipendenti (Nome,Cognome,Sesso,matricola,qualifica) VALUES ('Andrea','Giannoni','F',25685,'infermiere');
```

```
ALTER TABLE paziente ADD  
COLUMN MedicoID INT;
```

```
ALTER TABLE paziente ADD  
FOREIGN KEY (MedicoID) REFERENCES dipendenti(employeeID) ON DELETE SET NULL;
```



ON DELETE SET NULL

ON DELETE SET NULL → quando la riga della tabella di riferimento (References) viene cancellata, la corrispondente FOREIGN KEY diventa NULL

DELETE FROM dipendenti WHERE employeeID=1;

patientID	Nome	Cognome	Sesso	DataNascita	cf	MedicoID
1	Anna	Rossi	F	1962-03-11	RSSNNA62C51L872P	NULL
2	Roberto	Marzio	M	1971-04-25	MRZRRT71D04F251R	NULL
3	Giampiero	Di Nicola	M	1937-07-03	DNCGPR37L07H821Q	NULL
4	Luciana	Nunziatella	F	1955-05-11	NNZLCN55R45F205N	NULL
5	Arianna	Lucchini	F	1982-02-02	LCCRNN82B42N127C	NULL
6	MariaLuisa	Gichetti	F	1953-08-25	GCTMLS53H65H832F	NULL



ON DELETE SET CASCADE

ON DELETE CASCADE → quando la riga della tabella di riferimento (References) viene cancellata, la corrispondente riga che contiene la primary key cancellata come FOREIGN KEY viene cancellata

DELETE FROM paziente WHERE patientID=5;

therapyID	patientID	Nome	CasaProduttrice	costo	PrincipioAttivo	DataInizio	DataFine
1	3	Cumadin	Bayer	45.00	warfarin	2003-07-15	NULL
2	1	Corvel	Dompe	3.00	carvedilolo	2015-01-17	2019-03-15
3	2	Enapren	Zambon	5.00	enelapril	1999-10-11	2009-11-03
5	4	Cumadin	Bayer	45.00	warfarin	2000-04-27	2010-10-23
6	2	Cumadin	Bayer	45.00	warfarin	2009-11-04	2019-09-21

Non ci sono più le righe in cui compariva il paziente con patientID = 5



Queries 2.4, 2.5 e 2.6

- 2.4 Conoscere tutte le terapie associate ad ogni paziente;
- 2.5 Conoscere cognome e nome dei pazienti trattati con aspirina;
- 2.6 Conoscere cognome, nome, data di nascita, terapia per quei pazienti la cui terapia è finita prima dell'anno 1992.



Queries 2.4, 2.5 e 2.6

definiamo la Vista accessoria $\text{Paz_Ter} \leftarrow \sigma_{(P.ID=T.ID)} \text{Paziente } P \times \text{Terapia } T$

2.4 Conoscere tutte le terapie associate ad ogni paziente;

$$2.4 \leftarrow \pi_{(\text{Cognome}, P.\text{Nome}, P.\text{Data_di_n}, T.\text{Nome})} \text{Paz_Ter}$$

2.5 Conoscere cognome e nome dei pazienti trattati con cumadin;

$$2.5 \leftarrow \pi_{(\text{Cognome}, P.\text{Nome})} \sigma_{(T.\text{Nome}=\text{'cumadin'})} \text{Paz_Ter}$$

2.6 Conoscere cognome, nome, data di nascita, terapia per quei pazienti la cui terapia è finita prima dell'anno 2019.

$$2.6 \leftarrow \pi_{(\text{Cognome}, P.\text{Nome}, \text{Data_di_n}, T.\text{Nome})} \sigma_{(\text{year}(\text{Data_Fine}) < 2019)} \text{Paz_Ter}$$



PRODOTTO CARTESIANO

```
SELECT *  
FROM paziente,terapia;
```

Errore nella proiezione (JOIN NATURALE campo sovrascritto)

patientID	Nome	Cognome	Sesso	DataNascita	cf	therapyID	CasaProduttrice	costo	PrincipioAttivo	Data
3	Cumadin	Rossi	F	1962-11-03	RSSNNA62C51L872P	1	Bayer	45.00	warfarin	2003
3	Cumadin	Marzio	M	1971-04-25	MRZRRT71D04F251R	1	Bayer	45.00	warfarin	2003
3	Cumadin	Di Nicola	M	1937-07-03	DNCGPR37L07H821Q	1	Bayer	45.00	warfarin	2003
3	Cumadin	Nunziatella	F	1955-05-11	NNZLCN55R45F205N	1	Bayer	45.00	warfarin	2003
3	Cumadin	Lucchini	F	1982-02-02	LCCRNN82B42N127C	1	Bayer	45.00	warfarin	2003
1	Corvel	Rossi	F	1962-11-03	RSSNNA62C51L872P	2	Dompe	3.00	carvedilolo	2015
1	Corvel	Marzio	M	1971-04-25	MRZRRT71D04F251R	2	Dompe	3.00	carvedilolo	2015
1	Corvel	Di Nicola	M	1937-07-03	DNCGPR37L07H821Q	2	Dompe	3.00	carvedilolo	2015
1	Corvel	Nunziatella	F	1955-05-11	NNZLCN55R45F205N	2	Dompe	3.00	carvedilolo	2015
1	Corvel	Lucchini	F	1982-02-02	LCCRNN82B42N127C	2	Dompe	3.00	carvedilolo	2015
2	Enapren	Rossi	F	1962-11-03	RSSNNA62C51L872P	3	Zambon	5.00	enelapril	1995
2	Enapren	Marzio	M	1971-04-25	MRZRRT71D04F251R	3	Zambon	5.00	enelapril	1995
2	Enapren	Di Nicola	M	1937-07-03	DNCGPR37L07H821Q	3	Zambon	5.00	enelapril	1995
2	Enapren	Nunziatella	F	1955-05-11	NNZLCN55R45F205N	3	Zambon	5.00	enelapril	1995
2	Enapren	Lucchini	F	1982-02-02	LCCRNN82B42N127C	3	Zambon	5.00	enelapril	1995
5	Corvel	Rossi	F	1962-11-03	RSSNNA62C51L872P	4	Dompe	3.00	carvedilolo	2007
5	Corvel	Marzio	M	1971-04-25	MRZRRT71D04F251R	4	Dompe	3.00	carvedilolo	2007
5	Corvel	Di Nicola	M	1937-07-03	DNCGPR37L07H821Q	4	Dompe	3.00	carvedilolo	2007
5	Corvel	Nunziatella	F	1955-05-11	NNZLCN55R45F205N	4	Dompe	3.00	carvedilolo	2007
5	Corvel	Lucchini	F	1982-02-02	LCCRNN82B42N127C	4	Dompe	3.00	carvedilolo	2007
4	Cumadin	Rossi	F	1962-11-03	RSSNNA62C51L872P	5	Bayer	45.00	warfarin	2003
4	Cumadin	Marzio	M	1971-04-25	MRZRRT71D04F251R	5	Bayer	45.00	warfarin	2003



JOIN

```
SELECT *  
FROM paziente JOIN terapia ON paziente.patientID = terapia.patientID;
```

TABELLE DI CALCOLO

CONDIZIONE DI JOIN

Success

6 rows

4:42 PM

0.714 seconds

Errore nella proiezione (JOIN
NATURALE, campo sovrascritto)

Explore	SQL	Data	Chart	Export ▾	🔗
---------	-----	------	-------	----------	---

patientID	Nome	Cognome	Sesso	DataNascita	cf	therapyID	CasaProduttrice	costo
3	Cumadin	Di Nicola	M	1937-07-03	DNCGPR37L07H821Q	1	Bayer	45.00
1	Corvel	Rossi	F	1962-11-03	RSSNNA62C51L872P	2	Dompe	3.00
2	Enapren	Marzio	M	1971-04-25	MRZRRT71D04F251R	3	Zambon	5.00
5	Corvel	Lucchini	F	1982-02-02	LCCRNN82B42N127C	4	Dompe	3.00
4	Cumadin	Nunziatella	F	1955-05-11	NNZLCN55R45F205N	5	Bayer	45.00
2	Cumadin	Marzio	M	1971-04-25	MRZRRT71D04F251R	6	Bayer	45.00



TIPI DI JOIN

- **inner** (join interno fra le due tabelle) corrisponde al theta-join del modello relazionale, dove la condizione viene espressa in `Condizione_di_join`;
- **left (outer)** (join esterno sinistro) viene valutato, con la condizione espressa in `Condizione_di_join`, il join interno sulle due tabelle, e tale risultato è arricchito con le righe della tabella di sinistra che non hanno righe nella tabella di destra (aggiunte righe a destra);
- **right (outer)** (join esterno destro) viene valutato, con la condizione espressa in `Condizione_di_join`, il join interno sulle due tabelle, e tale risultato è arricchito con le righe della tabella di destra che non hanno righe nella tabella di sinistra (aggiunte righe a sinistra);
- **full (outer)** (join esterno completo) viene valutato, con la condizione espressa in `Condizione_di_join`, il join interno sulle due tabelle, e tale risultato è arricchito con le righe di entrambe le tabelle che non hanno righe corrispondenti nell'altra;



Query 2.4 : implementazione

```
SELECT paziente.Nome AS Nome, paziente.cognome AS Cognome, cf AS  
CodiceFiscale,terapia.Nome AS terapia
```

```
FROM paziente JOIN terapia ON paziente.patientID = terapia.patientID;
```

Success

6 rows

Explore

SQL

Data

Chart

Export ▾



Nome	Cognome	CodiceFiscale	terapia
Anna	Rossi	RSSNNA62C51L872P	Corvel
Roberto	Marzio	MRZRRT71D04F251R	Enapren
Roberto	Marzio	MRZRRT71D04F251R	Cumadin
Giampiero	Di Nicola	DNCGPR37L07H821Q	Cumadin
Luciana	Nunziatella	NNZLCN55R45F205N	Cumadin
Arianna	Lucchini	LCCRNN82B42N127C	Corvel



ESEMPIO: LEFT JOIN

```
SELECT paziente.Nome AS Nome,  
Cognome, Sesso, terapia.Nome AS  
terapia
```

```
FROM paziente JOIN terapia ON  
paziente.patientID =  
terapia.patientID;
```

```
SELECT paziente.Nome AS Nome,  
Cognome, Sesso, terapia.Nome AS  
terapia
```

```
FROM paziente LEFT JOIN terapia  
ON paziente.patientID =  
terapia.patientID;
```

Nome	Cognome	Sesso	terapia
Anna	Rossi	F	Corvel
Roberto	Marzio	M	Enapren
Roberto	Marzio	M	Cumadin
Giampiero	Di Nicola	M	Cumadin
Luciana	Nunziatella	F	Cumadin
Arianna	Lucchini	F	Corvel
Arianna	Lucchini	F	Nebilox

Nome	Cognome	Sesso	terapia
Anna	Rossi	F	Corvel
Roberto	Marzio	M	Enapren
Roberto	Marzio	M	Cumadin
Giampiero	Di Nicola	M	Cumadin
Luciana	Nunziatella	F	Cumadin
Arianna	Lucchini	F	Corvel
Arianna	Lucchini	F	Nebilox
MariaLuisa	Gichetti	F	NULL



Query 2.5 : implementazione

```
SELECT paziente.Nome AS Nome, paziente.cognome AS Cognome  
FROM paziente JOIN terapia ON paziente.patientID = terapia.patientID  
WHERE terapia.Nome='Cumadin';
```

Success

3 rows

Nome	Cognome
Giampiero	Di Nicola
Luciana	Nunziatella
Roberto	Marzio



Query 2.6 : implementazione

```
SELECT paziente.Nome AS Nome, paziente.cognome AS Cognome, DataNascita,  
terapia.nome AS terapia
```

```
FROM paziente JOIN terapia ON paziente.patientID = terapia.patientID
```

```
WHERE YEAR(terapia.DataFine)<2019;
```

Success

3 rows

Nome	Cognome	DataNascita	terapia
Roberto	Marzio	1971-04-25	Enapren
Luciana	Nunziatella	1955-05-11	Cumadin
Arianna	Lucchini	1982-02-02	Corvel



Queries 2.7, 2.8 e 2.9

2.7 Conoscere nome e cognome dei pazienti che hanno avuto terapie con inizio nell'anno 2000 o nell'anno 2007, senza considerare in quest'ultimo caso i pazienti nati prima del 1985;

2.8 Conoscere nome e cognome dei pazienti, senza considerare quelli nati prima del 1982, che hanno avuto terapie con inizio nell'anno 2000 o nell'anno 2007;

2.9 Conoscere cognome e nome dei pazienti che hanno terapie in corso.



Queries 2.7, 2.8 e 2.9

2.7 Conoscere nome e cognome dei pazienti che hanno avuto terapie con inizio nell'anno 2000 o nell'anno 2007, senza considerare in quest'ultimo caso i pazienti nati prima del 1985;

$$2.7 \leftarrow \pi_{(\text{Cognome}, \text{P.Nome})} \sigma_{([\text{year}(\text{Data_Inizio})=2000] \text{ OR } [\text{year}(\text{Data_Inizio})=2007 \text{ AND } \text{year}(\text{Data_di_n}) \geq 1985])} \text{Paz_Ter}$$

2.8 Conoscere nome e cognome dei pazienti, senza considerare quelli nati prima del 1982, che hanno avuto terapie con inizio nell'anno 2000 o nell'anno 2007;

$$2.8 \leftarrow \pi_{(\text{Cognome}, \text{P.Nome})} \sigma_{([\text{year}(\text{Data_di_n}) \geq 1982] \text{ AND } [\text{year}(\text{Data_Inizio})=2000] \text{ OR } [\text{year}(\text{Data_Inizio})=2007])} \text{Paz_Ter}$$

2.9 Conoscere cognome e nome dei pazienti che hanno terapie in corso.

$$2.9 \leftarrow \pi_{(\text{Cognome}, \text{P.Nome})} \sigma_{(\text{Data_Fine}=\text{NULL})} \text{Paz_Ter}$$



Query 2.7 : implementazione

```
SELECT paziente.Nome AS Nome, paziente.cognome AS Cognome  
FROM paziente JOIN terapia ON paziente.patientID = terapia.patientID  
WHERE YEAR(terapia.DataInizio)=2000 OR (YEAR(terapia.DataInizio)=2007 AND  
YEAR(paziente.DataNascita)>=1985);
```

Success

1 rows

Nome	Cognome
Luciana	Nunziatella



Query 2.8 : implementazione

```
SELECT paziente.Nome AS Nome, paziente.cognome AS Cognome  
FROM paziente JOIN terapia ON paziente.patientID = terapia.patientID  
WHERE (YEAR(terapia.DataInizio)=2000 OR YEAR(terapia.DataInizio)=2007) AND  
YEAR(paziente.DataNascita)>=1982;
```

Success

1 rows

Nome	Cognome
Arianna	Lucchini



Query 2.9 : implementazione

```
SELECT paziente.Nome AS Nome, paziente.cognome AS Cognome  
FROM paziente JOIN terapia ON paziente.patientID = terapia.patientID  
WHERE terapia.DataFine IS NULL;
```

Success

1 rows

Nome	Cognome
Giampiero	Di Nicola



COSTRUTTI E OPERATORI

- SELECT DISTINCT – permette di visualizzare solo le tuple non duplicate
- COUNT (NomeAtt) – riporta come risultato il numero di tuple che soddisfano la condizione, contando i valori dell'attributo
- [SUM, AVG, MIN, MAX](NomeAtt) – riportano come risultato il valore dell'operazione algebrica di riferimento sulla Colonna indicata
- CONDIZIONE SU OPERATORI AGGREGATI – HAVING – permette di usare gli operatori aggregati nella clausola WHERE



Queries 2.10, 2.11, e 2.12

2.10 Conoscere cognome e nome dei pazienti che hanno terapie iniziate prima del 2015, ma senza duplicati;

2.11 Conoscere la media del costo di tutte le terapie associate ad ogni paziente in corso dal 2017;

2.12 Conoscere nome, cognome e numero di terapie associate ad ogni paziente;

2.12bis Conoscere nome, cognome e numero di terapie associate ad ogni paziente considerando solo i pazienti con più di una terapia

Query 2.10 : implementazione

```
SELECT paziente.Nome AS Nome, paziente.cognome AS Cognome  
FROM paziente JOIN terapia ON paziente.patientID = terapia.patientID  
WHERE (YEAR(terapia.DataInizio)<2015);
```

Success

5 rows

Nome	Cognome
Roberto	Marzio
Roberto	Marzio
Giampiero	Di Nicola
Luciana	Nunziatella
Arianna	Lucchini

```
SELECT DISTINCT paziente.Nome AS Nome,  
paziente.cognome AS Cognome  
FROM paziente JOIN terapia ON paziente.patientID =  
terapia.patientID  
WHERE (YEAR(terapia.DataInizio)<2015);
```

Success

4 rows

Nome	Cognome
Roberto	Marzio
Giampiero	Di Nicola
Luciana	Nunziatella
Arianna	Lucchini



Query 2.11 : implementazione

```
SELECT AVG(costo) AS CostoMedio  
FROM terapia JOIN paziente ON paziente.patientID = terapia.patientID  
WHERE (YEAR(DataInizio)>=2017 OR YEAR(DataFine) IS NULL OR  
YEAR(DataFine)>=2017);
```

CostoMedio

20.600000

```
SELECT AVG(costo) AS CostoMedio,paziente.Cognome AS Cognome  
FROM terapia JOIN paziente ON paziente.patientID = terapia.patientID  
WHERE (YEAR(DataInizio)>=2017 OR YEAR(DataFine) IS NULL OR  
YEAR(DataFine)>=2017)  
GROUP BY(paziente.patientID);
```

CostoMedio	Cognome
------------	---------

3.000000	Rossi
----------	-------

45.000000	Marzio
-----------	--------

45.000000	Di Nicola
-----------	-----------

5.000000	Lucchini
----------	----------



Query 2.12 : implementazione

```
SELECT COUNT(terapia.PatientID) AS NumTerapie, paziente.Nome AS Nome,  
paziente.Cognome AS Cognome
```

```
FROM terapia JOIN paziente ON paziente.patientID = terapia.patientID
```

```
GROUP BY (terapia.PatientID);
```

NumTerapie	Nome	Cognome
1	Anna	Rossi
2	Roberto	Marzio
1	Giampiero	Di Nicola
1	Luciana	Nunziatella
2	Arianna	Lucchini



Query 2.12bis : implementazione

```
SELECT COUNT(terapia.PatientID) AS NumTerapie, paziente.Nome AS  
Nome, paziente.Cognome AS Cognome
```

```
FROM terapia JOIN paziente ON paziente.patientID = terapia.patientID
```

```
GROUP BY (terapia.PatientID)
```

```
HAVING NumTerapie>1;
```

NumTerapie	Nome	Cognome
2	Roberto	Marzio
2	Arianna	Lucchini



Esercizi: Queries 2.13, 2.14 e 2.15

- 2.13 Conoscere cognome, nome di quei pazienti che hanno avuto più terapie concomitanti;
- 2.14 Ottenere l'elenco ordinato alfabeticamente per cognome dei pazienti considerati nella tabella "Paziente";
- 2.15 Ottenere il numero dei pazienti con almeno una terapia;



Esercizi: Queries 2.16, 2.17, 2.18

- 2.16 Ottenere il numero delle terapie registrate nella base di dati, considerando la sola tabella “Terapia”;

- 2.17 Ottenere l’indicazione, per ogni paziente con almeno una terapia, del numero di terapie e del codice;

- 2.18 Ottenere l’indicazione, per ogni paziente con terapie, del numero di terapie e dei dati anagrafici del paziente;



OPERATORE LIKE

- Definizione di pattern all'interno di stringhe:
 - % = qualsiasi numero di caratteri
 - _ = un solo carattere
- Parola chiave nella clausola WHERE: LIKE al posto di =

```
SELECT *  
FROM terapia  
WHERE PrincipioAttivo LIKE '%o';
```

Queries 2.20 e 2.21



- 2.20 ottenere i nomi, cognomi e terapie associate ad ogni paziente che assuma beta-bloccanti (desinenza -lolo)
- 2.21 conoscere tutti i dati dei pazienti nati in luglio



Query 2.20: implementazione

```
SELECT paziente.Nome, paziente.Cognome, terapia.Nome AS Terapia  
FROM paziente JOIN terapia ON paziente.patientID=terapia.patientID  
WHERE PrincipioAttivo LIKE '%lolo';
```

Success

3 rows

Nome	Cognome	Terapia
Anna	Rossi	Corvel
Arianna	Lucchini	Corvel
Arianna	Lucchini	Nebilox



Query 2.21: implementazione

```
SELECT *
```

```
FROM paziente
```

```
WHERE DataNascita LIKE '____07____';
```

patientID	Nome	Cognome	Sesso	DataNascita	cf	MedicoID
3	Giampiero	Di Nicola	M	1937-07-03	DNCGPR37L07H821Q	NULL



OPERATORE UNION

- Operazione di unione insiemistica
- Vincoli:
 - Si possono unire due tabelle con lo STESSO NUMERO DI COLONNE
 - Le colonne selezionate devono AVERE LO STESSO DOMINIO

```
SELECT  
Nome,Cognome,Sesso  
FROM paziente
```

UNION

```
SELECT Nome, Cognome,  
Sesso  
FROM dipendenti;
```

Nome	Cognome	Sesso
Anna	Rossi	F
Roberto	Marzio	M
Giampiero	Di Nicola	M
Luciana	Nunziatella	F
Arianna	Lucchini	F
MariaLuisa	Gichetti	F
Bruno	Mascheroni	M
Anna	Trilli	F
Vincenzo	Cinti	M
MariaChiara	Bandai	F
Andrea	Giannoni	F



Query 2.22

- Conoscere nome, cognome e qualifica di tutte le persone presenti in ospedale



Query 2.22: implementazione

```
SELECT Nome,Cognome,cf AS qualifica
FROM paziente
UNION
SELECT Nome, Cognome, qualifica
FROM dipendenti;
```

La tabella paziente non ha
“qualifica” e quindi devo
selezionare un campo che sia
compatibile come dominio

Nome	Cognome	qualifica
Anna	Rossi	RSSNNA62C51L872P
Roberto	Marzio	MRZRRT71D04F251R
Giampiero	Di Nicola	DNCGPR37L07H821Q
Luciana	Nunziatella	NNZLCN55R45F205N
Arianna	Lucchini	LCCRNN82B42N127C
MariaLuisa	Gichetti	GCTMLS53H65H832F
Bruno	Mascheroni	dirigente medico
Anna	Trilli	dirigente medico
Vincenzo	Cinti	infermiere
MariaChiara	Bandai	tecnico radiologia
Andrea	Giannoni	infermiere



QUERY ANNIDATE

- È possibile usare il risultato di una query all'interno di un'altra query
- Il DBMS eseguirà prima la query più interna e poi quella più esterna
- Le query annidate vanno inserite tra parentesi

```
SELECT paziente.Nome, paziente.Cognome  
FROM paziente  
WHERE paziente.patientID IN  
(  
SELECT terapia.patientID  
FROM terapia  
WHERE PrincipioAttivo LIKE '%lolo');
```



ESEMPIO (1/2)

- Ottenere il Codice Fiscale di quei pazienti che hanno speso più di 25 euro per la terapia.
- STEP 1 – ottenere gli ID dei pazienti che hanno speso più di 25 euro

```
SELECT terapia.patientID, SUM(terapia.costo) AS importoTot  
FROM terapia  
GROUP BY terapia.patientID  
HAVING SUM(terapia.costo)>25;
```

patientID	importoTot
2	50.00
3	45.00
4	45.00

- STEP 2a – visualizzare il CF a partire da questa selezione

```
SELECT paziente.cf  
FROM paziente  
WHERE paziente.patientID IN  
(  
SELECT terapia.patientID AS pID, SUM(terapia.costo) AS importoTot  
FROM terapia  
GROUP BY terapia.patientID  
HAVING SUM(terapia.costo)>25);
```

NON FUNZIONA PERCHÈ LA
QUERY NESTED RITORNA 2
COLONNE. DEVO QUINDI
OPERARE UNA DOPPIA
SELEZIONA



ESEMPIO (2/2)

- Step 2b – seleziono la sola Colonna patientID e poi applico la condizione WHERE ... IN

```
SELECT paziente.cf  
FROM paziente  
WHERE paziente.patientID IN (  
SELECT pID  
FROM  
(  
SELECT terapia.patientID AS  
pID,SUM(terapia.costo) AS importoTot  
FROM terapia  
GROUP BY terapia.patientID  
HAVING SUM(terapia.costo)>25) AS NewTer);
```

cf

DNCGPR37L07H821Q

MRZRRT71D04F251R

NNZLCN55R45F205N

DEFINIRE SEMPRE L'ALIAS