

The background features a large orange shape with a white curved cutout on the right. Inside the orange area are several short, yellow dashed lines and a single solid yellow circle.

# Corso di Alta Formazione

Data Science

Metodi Informatici per lo Sviluppo e Gestione di Databases

Luigi Corsaro

# Presentazioni

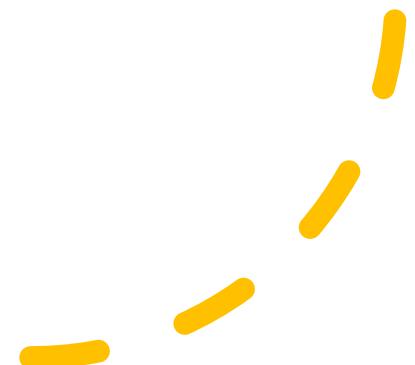
Chi siamo

Esperienza con database



# Scopo del corso

- Operare con un database al fine di saper archiviare, gestire e consultare i dati
  - Introduzione
  - Concetti base architettura client server
  - Strumenti Necessari
  - Creazione di un database
  - Importazione dei dati
  - Progettazione di un data base
  - Manipolazione dei dati
  - Esportazione dei dati
  - Salvataggio e ripristino



Cosa ci  
portiamo a  
casa

Concetti

Strumenti

Esperienza pratica

# Introduzione: chi lavora con i databases



**Progettista di Database**



**Data Scientist**



**Database Administrator**



**Analista/Sviluppatore SQL**



**Database Specialist**



**Business Intelligence Specialist**

Quali conoscenze si devono avere per utilizzare al meglio i database

<b>Area di Competenza</b>	<b>Livello</b>
Dominio	Elevato
SQL	Medio - Elevato
Progettazione RDBMS	Base
Sistema Operativo	Base
Architettura Informatica	Base

# Perché lavorare con i dati



Sono un «bene» aziendale allo stesso livello del personale, immobili, macchinari

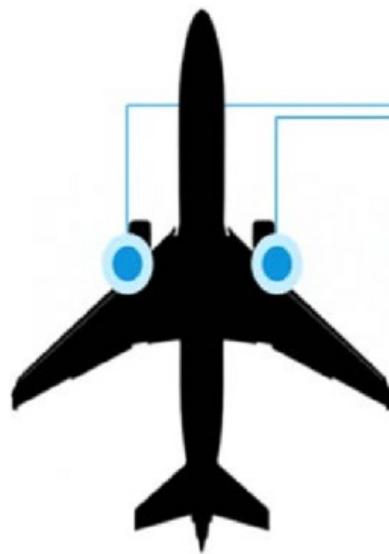


Sono un prodotto dell'azienda che rappresentano l'azienda



DATI = informazione =  
conoscenza = base delle  
decisioni!

Quantità di  
dati generata  
in volo  
nazionale US



$$20 \text{ TB} \times 2 \times 6 \times 28,537 \times 365$$

20 terabytes of  
information per  
engine every hour

twin-engine  
Boeing 737

six-hour, cross-  
country flight from  
New York to Los  
Angeles

# of commercial  
flights in the sky in  
the United States on  
any given day.

days in a year

$$= 2,499,841,200 \text{ TB}$$

# Tipi di DBMS

Gerarchico

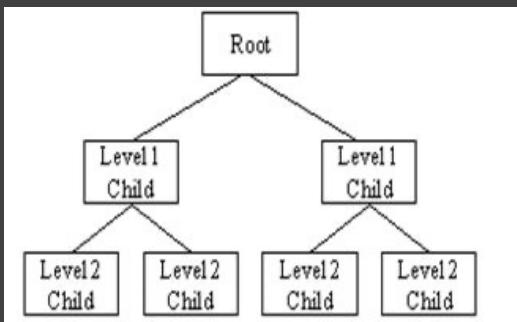
Reticolare

Relazionale

Orientato ad oggetti

Orientati ai documenti

# Gerarchico



Anni 60



Accesso per la lettura estremamente veloce, struttura chiara, semplice a livello tecnico



Struttura ad albero rigida che non consente collegamenti tra gli alberi



Banche, assicurazioni, sistemi operativi



IMS/DB

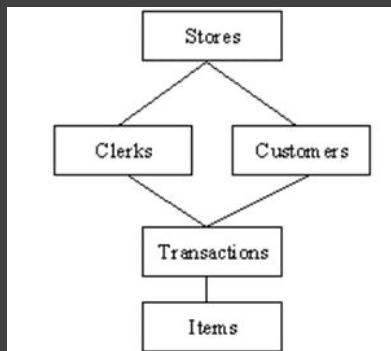


Esempio Moderno: XML

# Esempio XML

```
<?xml version="1.0" encoding="UTF-8"?>
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
<CATALOG>
```

# Reticolare



Inizio anni 70

Numero maggiore di percorsi per accedere al record, non c'è una rigida gerarchia

Panoramica scarsa per database di grandi dimensioni

Grandi calcolatori

UDS (Siemens), DMS (Sperry Univac)

# Relazionale



1970



Creazione ed elaborazione facile e flessibile, facile espansione, facile da implementare



Ingestibile per grandi quantità di dati, scarsa segmentazione, attributi artificiali delle chiavi, interfaccia di programmazione esterna, raffigurazione carente delle proprietà e comportamento degli oggetti



Controlling, contabilità, sistemi di gestione della merce, sistemi di gestione dei contenuti e molti altri.



MySQL, PostgreSQL, Oracle, SQLite, DB2, Ingres, MariaDB, Microsoft Access, MSSQL

# Orientato ad oggetti

Fine anni 80

Migliore supporto di linguaggi di programmazione orientati agli oggetti, memorizzazione di contenuti multimediali

Prestazioni peggiorano all'aumentare dei dati, poche interfacce compatibili

Inventari (musei, vendita al dettaglio)

db4o

# Orientati ai documenti

Anni 80

Archiviazione centrale dei dati correlati in documenti singoli, struttura libera, orientamento multimediale

Sforzi organizzativi relativamente elevati, spesso richiedono competenze di programmazione

Applicazioni web, motori di ricerca Internet, database di testo

Lotus Notes, Amazon SimpleDB, MongoDB, CouchDB, Riak, ThruDB, OrientDB

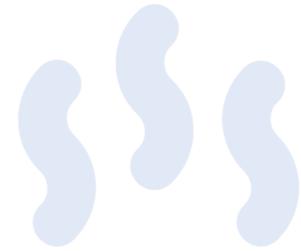


Excel

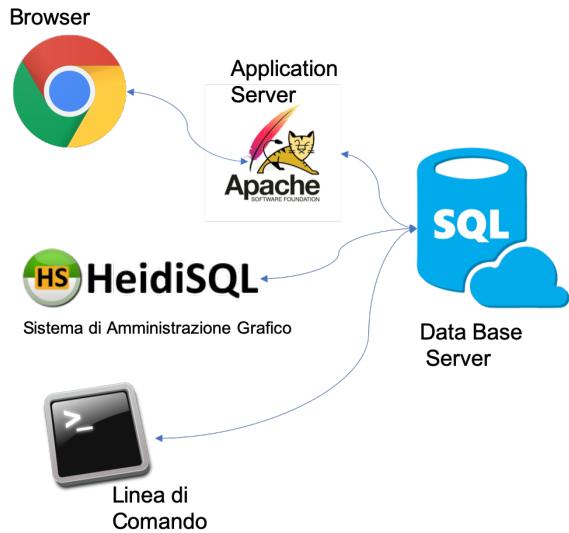




Break



# Architettura client server



Wikipedia: In [informatica](#) il termine **sistema client-server** (letteralmente *cliente-serviente*) indica un'[architettura di rete](#) nella quale genericamente un [computer client](#) o [terminale](#) si [connette](#) ad un [server](#) per la fruizione di un certo servizio, quale ad esempio la condivisione di una certa [risorsa hardware/software](#) con altri client, appoggiandosi alla sottostante architettura [protocollo](#).

# Architettura client server: Perché ?

Flusso dati:  
dove sono i  
dati e come si  
muovono

Modalità di  
accesso: chi  
accede a cosa  
e come

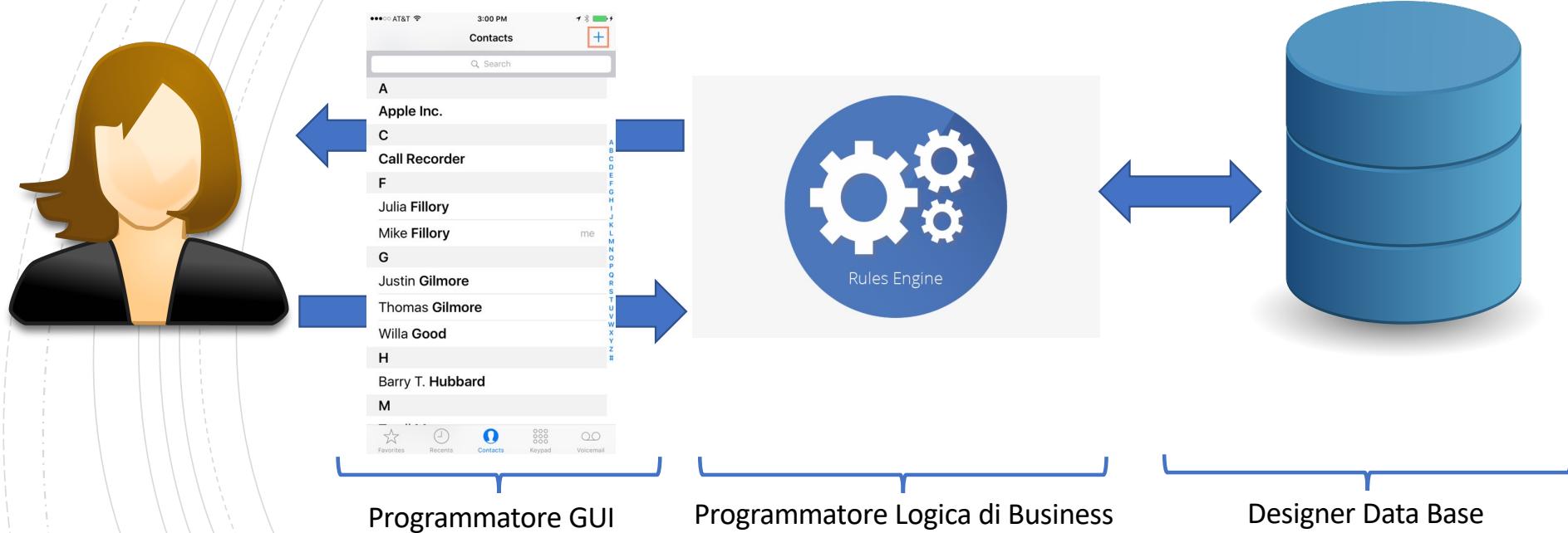
Performance:  
in test o in  
produzione

Sicurezza:  
non  
dimenticare



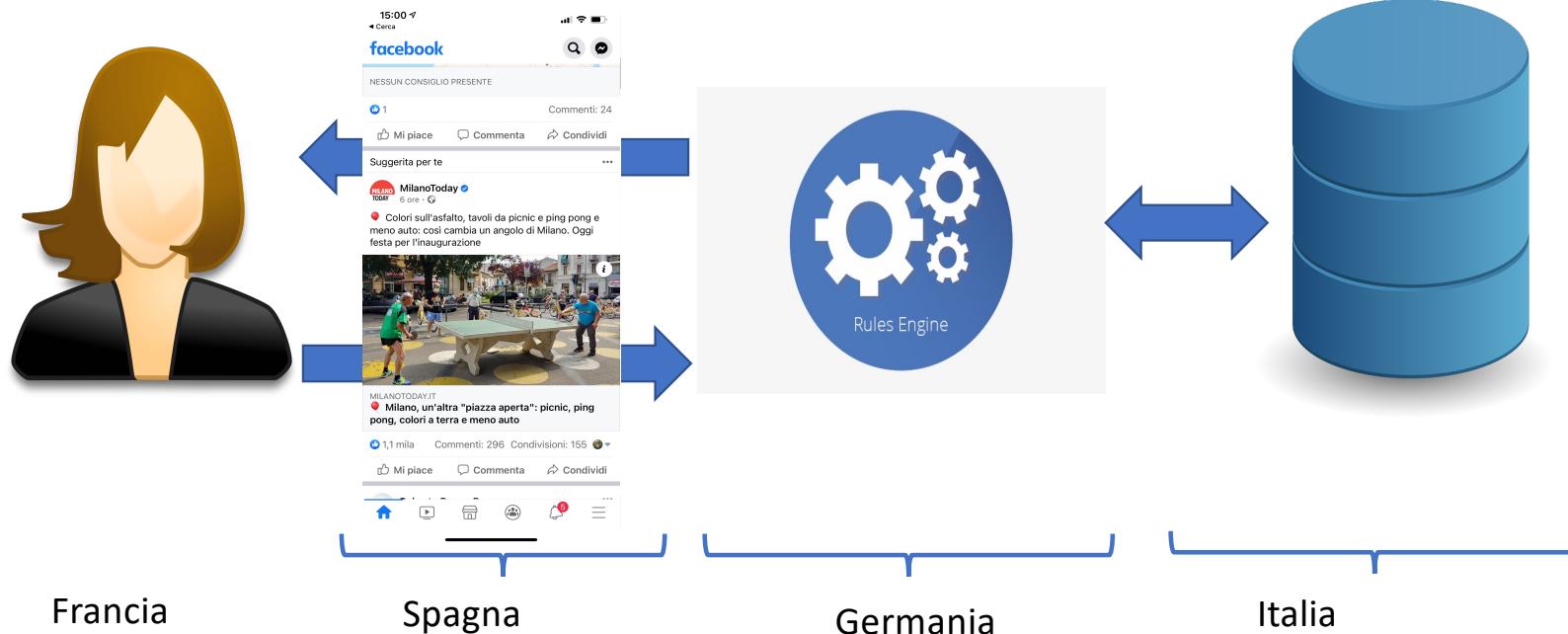
Chi usa i  
databases  
nella vita quotidiana...

# Rubrica telefonica



Tutto in unico dispositivo: il telefono !

# Facebook





MS Access  
FileMaker Pro

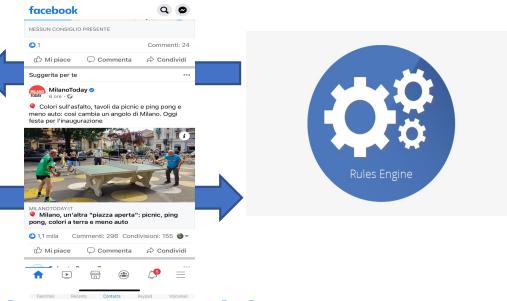


Come accediamo in  
questo corso ?

# phpmyadmin



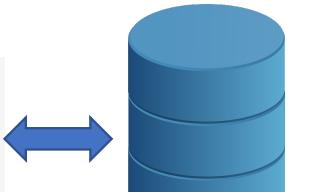
Francia



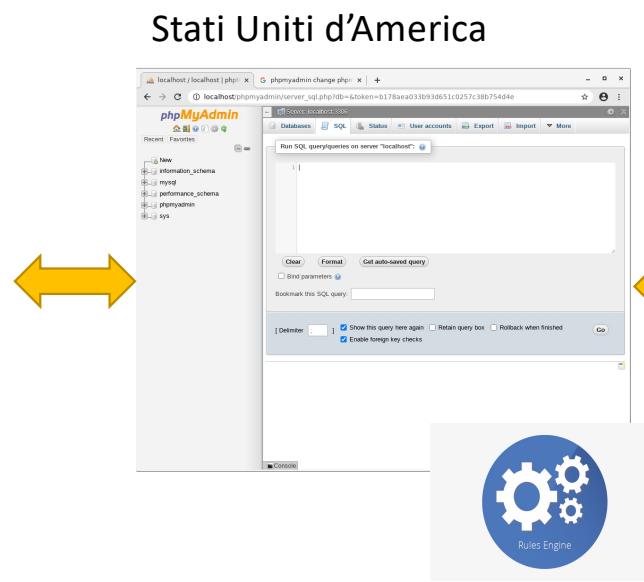
Spagna



Germania



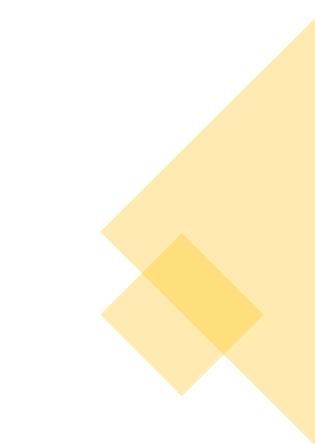
Italia



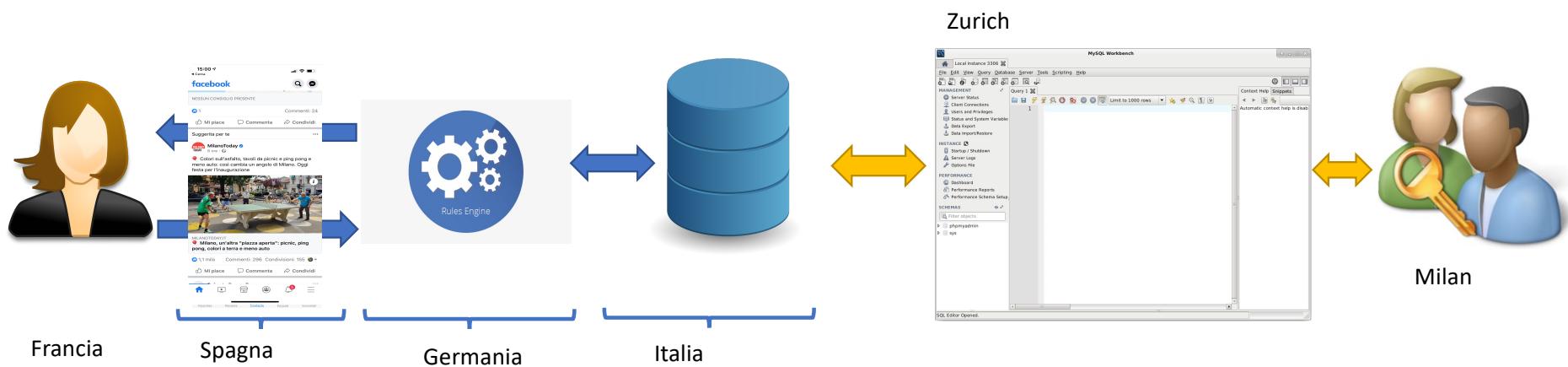
Web Server



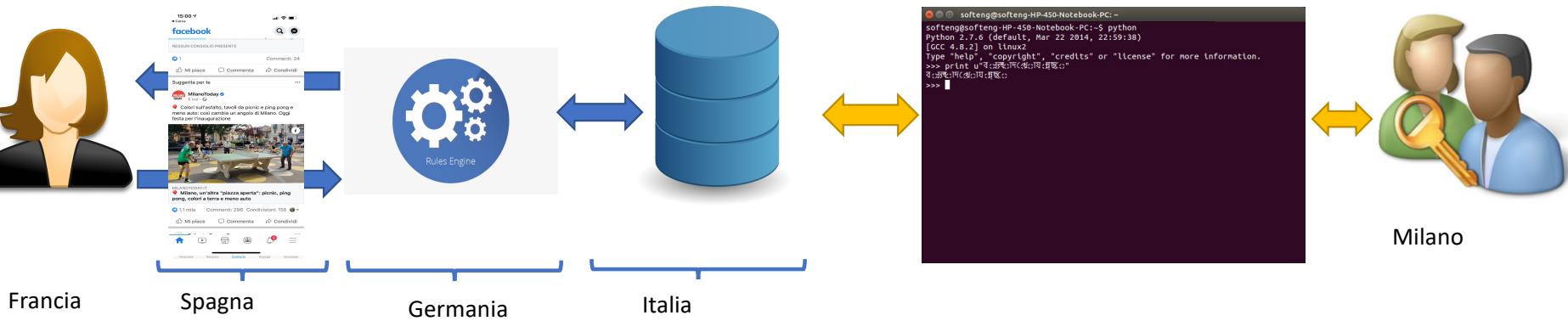
Stati Uniti d'America



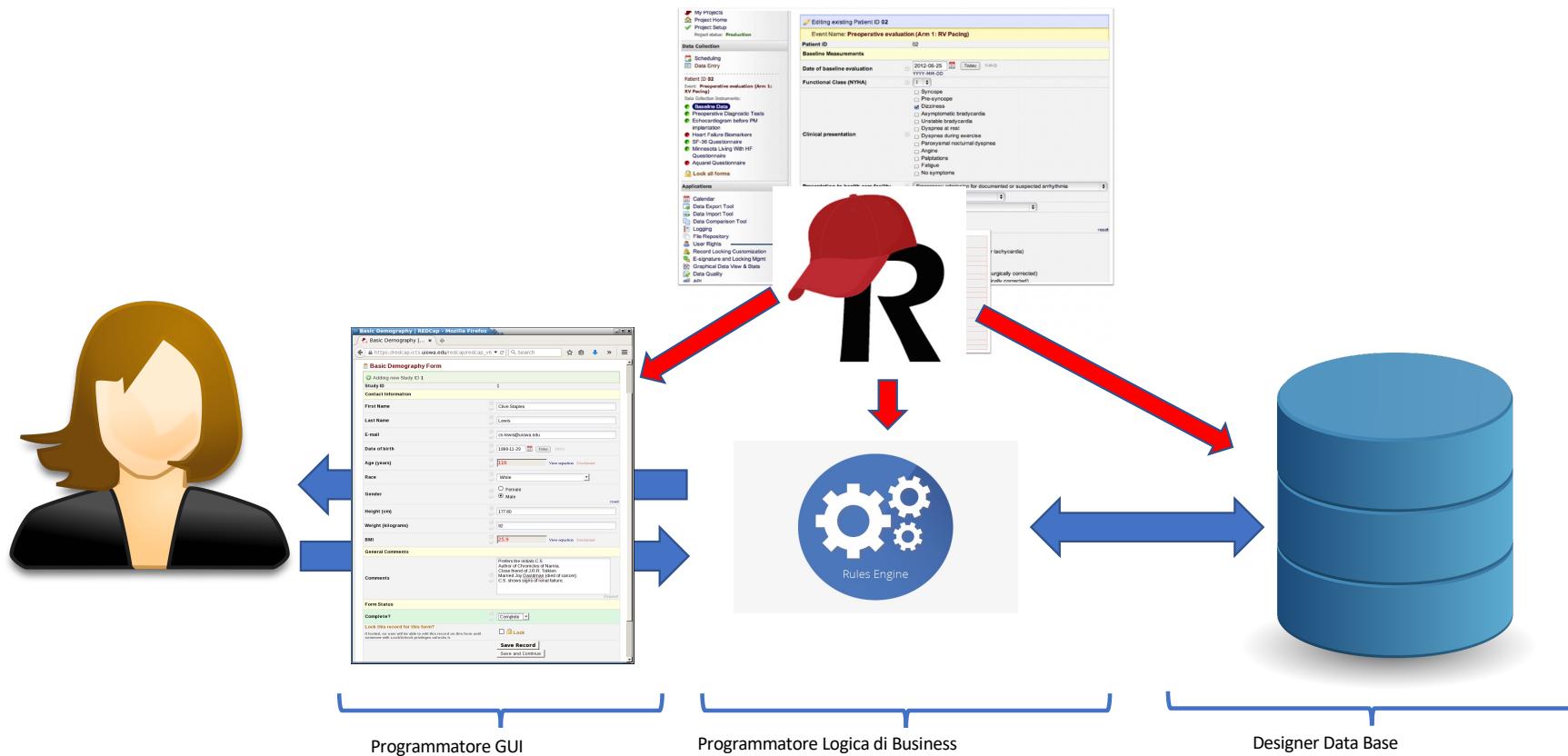
# MysqlWorkbench



# Shell



# redCap - metaprogramma



**Basic Demography | REDCap - Mozilla Firefox**

Adding new Study ID 1

Study ID 1

### Contact Information

First Name	Clive Staples
Last Name	Lewis
E-mail	cs-lewis@uiowa.edu
Date of birth	1898-11-29 <input type="button" value="Today"/>
Age (years)	116 <a href="#">View equation</a> <a href="#">Disclaimer</a>
Race	White
Gender	<input type="radio"/> Female <input checked="" type="radio"/> Male
Height (cm)	177.80
Weight (kilograms)	82
BMI	25.9 <a href="#">View equation</a> <a href="#">Disclaimer</a>

### General Comments

Comments

Prefers the initials C.S.  
Author of *Chronicles of Narnia*.  
Close friend of J.R.R. Tolkien.  
Married Joy Davidman (died of cancer).  
C.S. shows signs of renal failure.

Form Status

Complete?

Lock this record for this form?

If locked, no user will be able to edit this record on this form until someone with Lock/Unlock privileges unlocks it.

## Utente

**My Projects**

- [Project Home](#)
- [Project Setup](#)

Project status: Production

---

**Data Collection**

- [Scheduling](#)
- [Data Entry](#)

---

**Patient ID** 02

Event: Preoperative evaluation (Arm 1: RV Pacing)

Data Categories:

- Baseline Data
- Preoperative Diagnostic Tests
- Echocardiogram before PM implantation
- Heart Failure Biomarkers
- SF-36 Questionnaire
- Minnesota Living With HF Questionnaire
- Aquarel Questionnaire

[Lock all forms](#)

---

**Applications**

- [Calendar](#)
- [Data Export Tool](#)
- [Data Import Tool](#)
- [Data Comparison Tool](#)
- [Logging](#)
- [File Repository](#)
- [User Rights](#)
- [Record Locking Customization](#)
- [E-signature and Locking Mgmt](#)
- [Graphical Data View & Stats](#)
- [Data Quality](#)
- [API](#)

### Editing existing Patient ID 02

Event Name: Preoperative evaluation (Arm 1: RV Pacing)

Patient ID: 02

**Baseline Measurements**

Date of baseline evaluation: 2012-06-25 (YYYY-MM-DD)

Functional Class (NYHA): I

- Syncope
- Pre-syncope
- Dizziness
- Asymptomatic bradycardia
- Unstable bradycardia
- Dyspnea at rest
- Dyspnea during exercise
- Paroxysmal nocturnal dyspnea
- Angina
- Palpitations
- Fatigue
- No symptoms

**Clinical presentation**

Presentation to health care facility: Emergency admission for documented or suspected arrhythmia

Etiology: Degenerative

Underlying heart disease: None

Any comorbid condition?

No Access    Read Only    View Only    Edit

Basic User Rights	No Access	Read Only	View Only	Edit
Calendar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Data Export Tool	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Data Import Tool	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Data Comparison Tool	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Logging	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Data Repository	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
User Rights	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Data Access Groups	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Graphical Data View & Stats	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Data Quality	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
What is a Data Rule?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Report & Report Builder	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Project Design & Builder	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
API	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
What is the REDCap API?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Setting up rules for record locking and e-signatures:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Record Locking Customization	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
LookUp/Unlock Records	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

# Programmatore



Ensembl.org



## Strumenti

Remote Desktop

Terminale

phpMyAdmin

MySQLWorkbench

ed ovviamente tanti altri ....

Macchina Virtuale

WebServer



# Dati di accesso

Indirizzo Server:

aulamed5.unipv.it

>Password

Sql2022Corso



## Amministratore di Sistema

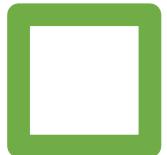
- Ha installato la macchina virtuale
- Ha installato MySql Server
- Ha creato gli utenti unix
- Ha creato gli utenti data base

# Sicurezza e Privacy

Utenti

Permessi

Crittografia



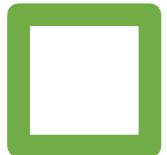
# / DB Administrator

	Create	Read	Update	Delete
Oggetti Server	Yes	Yes	Yes	Yes
Oggetti di uno schema	Yes	Yes	Yes	Yes
Dati di uno schema	Yes	Yes	Yes	Yes



# / DB Designer

	Create	Read	Update	Delete
Oggetti Server	No	No	No	No
Oggetti di uno schema	Yes	Yes	Yes	Yes
Dati di uno schema	Yes	Yes	Yes	Yes



# Utente

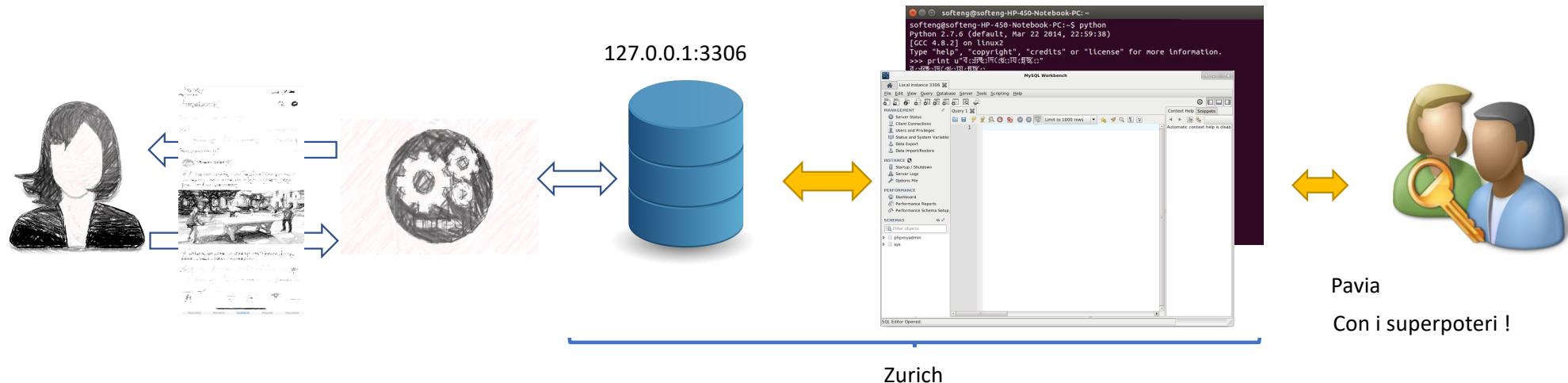


	Create	Read	Update	Delete
Oggetti Server	No	No	No	No
Oggetti di uno schema	No	No	No	No
Dati di uno schema	Yes	Yes	Yes	Yes

# Preparazione nostro PC

- Scarico MySql a seconda del mio sistema operativo
  - Google: mysql community server download
  - <https://dev.mysql.com/downloads/mysql/>
- Installo MySql
- Configurazione utente “root”
- Scarico ed Installo MySql Workbench
  - <https://dev.mysql.com/downloads/workbench/>
- Accesso
  - Terminale
  - MySql WorkBench

# La nostra architettura



# Accesso Console

## Accesso con terminale

- Accedo con Remote Desktop alla macchina virtuale
- Apro il terminale
- accedo a mysql server  
`mysql –u user_xxxx –h  
127.0.0.1 -p`

# Accesso Console

## Accesso con phpmyadmin

- Accedo con Remote Desktop alla macchina virtuale
- Avvio phpmyadmin
- Accedo

# Accesso Console

## Accesso con MySql Workbench

- Accedo con Remote Desktop alla macchina virtuale
- Avvio MySql Workbench
- Creo il collegamento



Come potremmo  
accedere dal nostro  
computer senza  
usare Remote  
Desktop

# Strutture dati

## Struttura

- Colonna è un attributo
- Tabella è una relazione
- Schema è un database

## Dati

- Record/Riga è una Tupla

# Set di caratteri e confronto

- Charset and collation:
  - I caratteri come sono rappresentati ?
  - Come li confronto ?
- <https://pinetools.com/sort-list>
  - òresund Bridge
  - Oristano
  - Arancia
  - ananas
  - Mela
  - mandarino
  - Àn

Case Sensitive – Case Insensitive ?  
Accent Sensitive - Accent Insensitive ?

# Creazione Data Base

Charset

Collation

Utente

# Creazione database da shell

---

```
mysql -u user_xxxx -p -h  
127.0.0.1
```

---

```
create database user_xxxx  
COLLATE 'utf8_ci';
```

---

```
show databases;
```

---

```
use user_xxxx;
```

---

Cancellazione  
database da  
shell

---

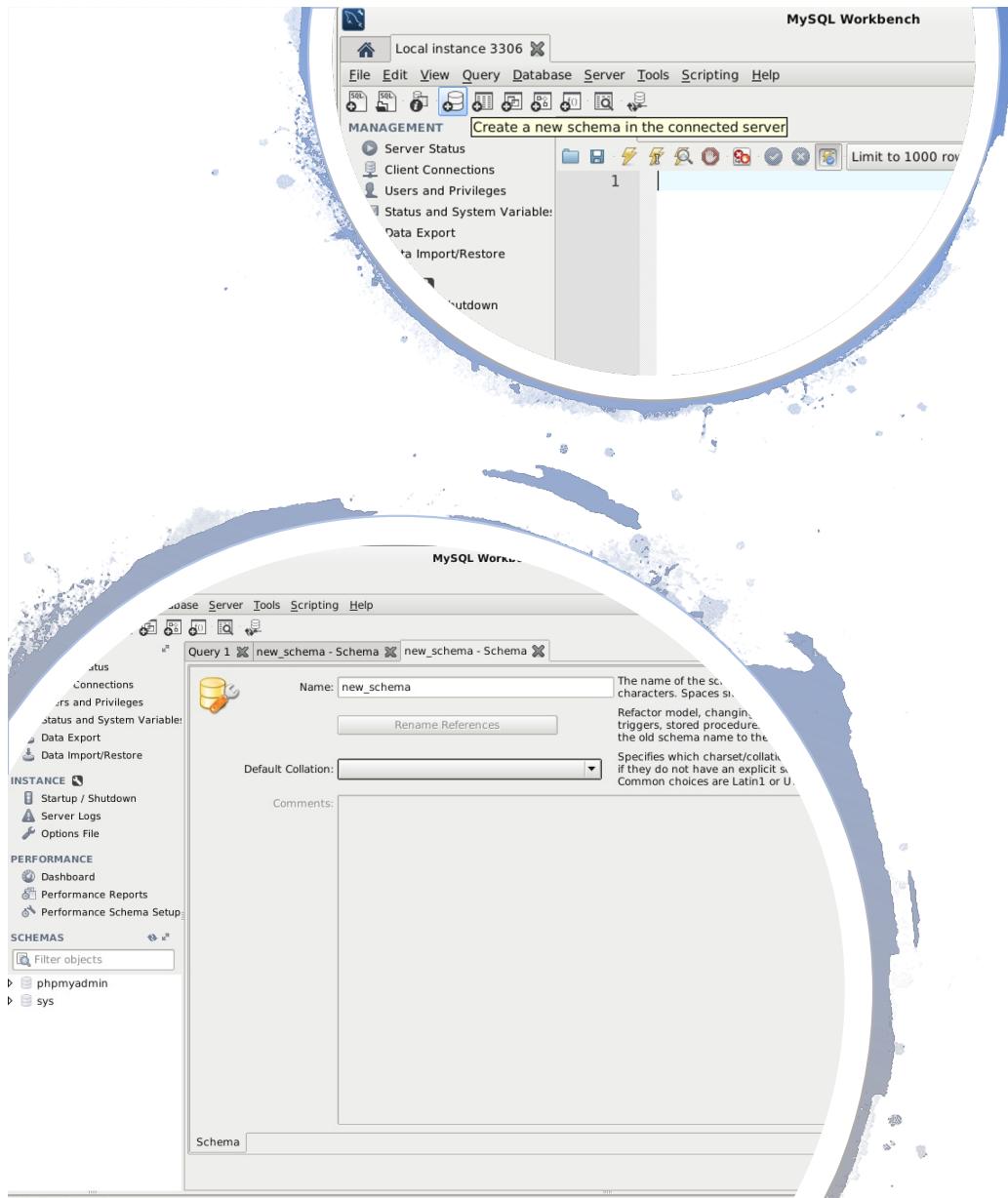
```
mysql -u user_xxxx  
-p -h 127.0.0.1
```

---

```
DROP DATABASE  
`user_xxxx`;
```

# Creazione Data Base con MySql Workbench

- Cliccare su icona crea database
- Nome DataBase: user\_xxxx
- Default Collation
- Apply



Cancellazione  
database

---

Tasto destro del  
mouse e Drop  
Schema

---

**DROP DATABASE  
user\_xxx;**

# Modifica e Cancellazione DataBase

Modifica

```
ALTER DATABASE user_xxxx COLLATE  
'latin2_bin';
```

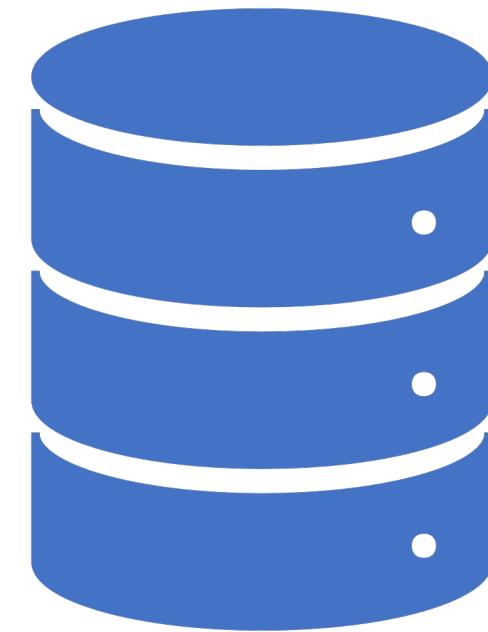
Rinomina

```
Non esiste il comando
```

Cancellazione

```
DROP DATABASE user_xxxx;
```

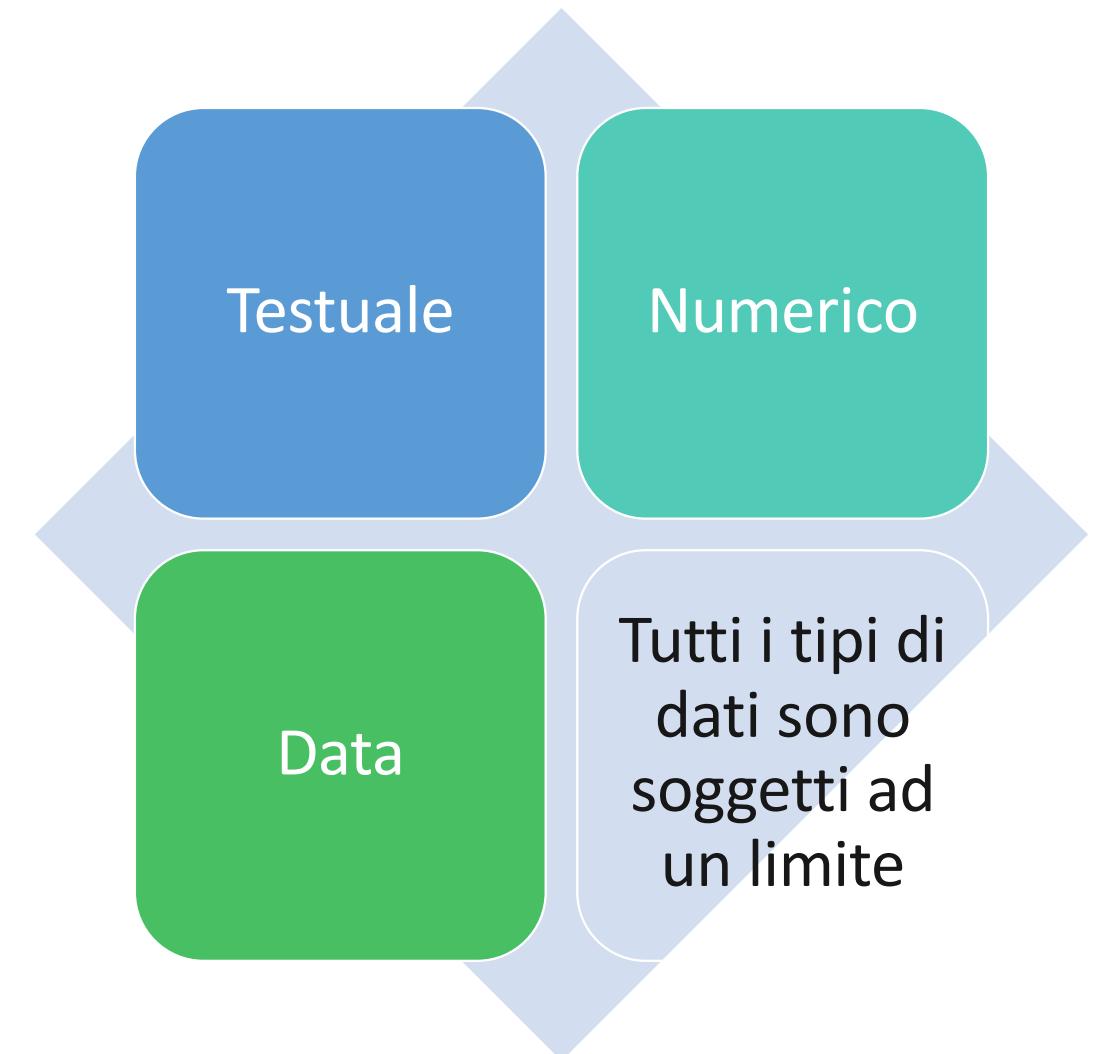
Ricreate il data  
base.





Cosa faccio per  
creare un  
database in  
Redcap ..

# Tipi di dati



Perché ?



ORDINAMENTO



PRECISIONE



DIMENSIONE



COERENZA



APPLICAZIONE  
DI FUNZIONI

# Tipo testo/binario

---

Data type	Description
CHAR(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters
VARCHAR(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. <b>Note:</b> If you put a greater value than 255 it will be converted to a TEXT type
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT	Holds a string with a maximum length of 65,535 bytes
BLOB	For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LONGBLOB	For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data
ENUM(x,y,z,etc.)	Let you enter a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted. <b>Note:</b> The values are sorted in the order you enter them. You enter the possible values in this format: ENUM('X','Y','Z')
SET	Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice

# Tipo numerico

---

Data type	Description
TINYINT(size)	-128 to 127 normal. 0 to 255 UNSIGNED*. The maximum number of digits may be specified in parenthesis
SMALLINT(size)	-32768 to 32767 normal. 0 to 65535 UNSIGNED*. The maximum number of digits may be specified in parenthesis
MEDIUMINT(size)	-8388608 to 8388607 normal. 0 to 16777215 UNSIGNED*. The maximum number of digits may be specified in parenthesis
INT(size)	-2147483648 to 2147483647 normal. 0 to 4294967295 UNSIGNED*. The maximum number of digits may be specified in parenthesis
BIGINT(size)	-9223372036854775808 to 9223372036854775807 normal. 0 to 18446744073709551615 UNSIGNED*. The maximum number of digits may be specified in parenthesis
FLOAT(size,d)	A small number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter
DOUBLE(size,d)	A large number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter
DECIMAL(size,d)	A DOUBLE stored as a string , allowing for a fixed decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter

# Tipo Data

---

Data type	Description
DATE()	A date. Format: YYYY-MM-DD <b>Note:</b> The supported range is from '1000-01-01' to '9999-12-31'
DATETIME()	*A date and time combination. Format: YYYY-MM-DD HH:MI:SS <b>Note:</b> The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'
TIMESTAMP()	*A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD HH:MI:SS <b>Note:</b> The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC
TIME()	A time. Format: HH:MI:SS <b>Note:</b> The supported range is from '-838:59:59' to '838:59:59'
YEAR()	A year in two-digit or four-digit format. <b>Note:</b> Values allowed in four-digit format: 1901 to 2155. Values allowed in two-digit format: 70 to 69, representing years from 1970 to 2069

# Nomi e Convenzioni

## Utilizzare sempre lo stesso stile

- Maiuscolo
- Minuscolo
- CamelCase
- Con separatore/Senza separatore

## Usare sempre lo stesso prefisso/suffisso

- ID
- Desc

## Creare nomi che dicano qualcosa

- Prefisso a\_ sta per archivio
- Ideale il nome singolare di cosa contiene la relazione: passaggio nel caso di un orario di passaggio di una linea di trasporto

# Perchè



Non saremo gli unici a lavorare sul database



La mia memoria è labile



Lavorare su un database di 100 tabelle complica le cose



Scrivere una «richiesta dati» senza una convenzione di denominazione è molto complicato

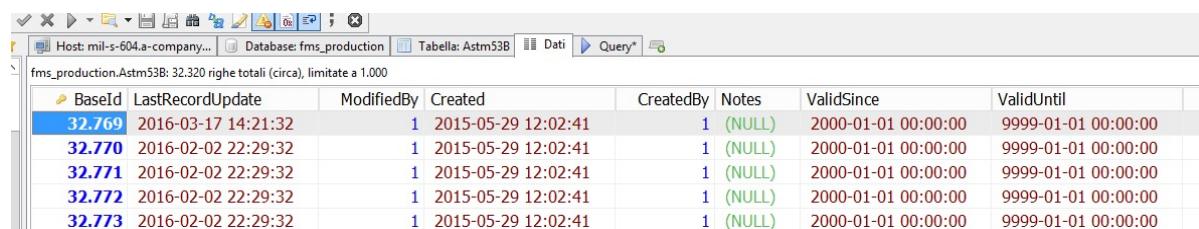


Una volta scelto il nome essere coerenti in tutto il database



DAY 2

# Tabella



The screenshot shows a database interface with a toolbar at the top. Below the toolbar, the connection details are displayed: Host: mil-s-604-a-company..., Database: fms\_production, Tabella: Astm53B. A message indicates there are 32.320 righe totali (circa), limitate a 1.000. The main area displays a table with the following data:

BaseId	LastRecordUpdate	ModifiedBy	Created	CreatedBy	Notes	ValidSince	ValidUntil
32.769	2016-03-17 14:21:32	1	2015-05-29 12:02:41	1	(NULL)	2000-01-01 00:00:00	9999-01-01 00:00:00
32.770	2016-02-02 22:29:32	1	2015-05-29 12:02:41	1	(NULL)	2000-01-01 00:00:00	9999-01-01 00:00:00
32.771	2016-02-02 22:29:32	1	2015-05-29 12:02:41	1	(NULL)	2000-01-01 00:00:00	9999-01-01 00:00:00
32.772	2016-02-02 22:29:32	1	2015-05-29 12:02:41	1	(NULL)	2000-01-01 00:00:00	9999-01-01 00:00:00
32.773	2016-02-02 22:29:32	1	2015-05-29 12:02:41	1	(NULL)	2000-01-01 00:00:00	9999-01-01 00:00:00

- Insieme organizzato di colonne (schema) e righe per la memorizzazione di informazioni (istanze)
- Colonne: detti campi o attributi della relazione, costituiscono lo schema
- Righe: dette anche record o tuple, ogni record è una istanza dello schema della tabella.

# Operazioni su tabella (relazione)



CREAZIONE  
SCHEMA RELAZIONE



LETTURA SCHEMA  
RELAZIONE



MODIFICA SCHEMA  
RELAZIONE



ELIMINAZIONE  
RELAZIONE



AGGIUNGERE UNA  
CHIAVE



AGGIUNGERE UN  
INDICE



# Preparazione

Creo il  
database

Seleziono il  
data base  
per l'utilizzo

# Selezione DataBase

Riga di comando:

- Per ogni comando user\_xxxx.oggetto oppure
- use user\_xxxx;
- .....

Grafica

- Doppio click sul mio database

# Creazione Tabella

The image shows two side-by-side windows of MySQL Workbench.

**Left Window (Schema Creation):**

- Title Bar:** MySQL Workbench - Local instance 3306
- Toolbar:** Standard MySQL Workbench toolbar.
- Management Panel:** Shows Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, and Data Import/Restore.
- INSTANCE Panel:** Shows Startup / Shutdown, Server Logs, and Options File.
- PERFORMANCE Panel:** Shows Dashboard, Performance Reports, and Performance Schema Setup.
- SCHEMAS Panel:** Shows phpmyadmin and sys. The sys schema is selected.
- Central Area:** A dialog titled "Create a new table in the active schema in connected server" is open. It contains fields for "Name:" (new\_schema), "Default Collation:" (empty dropdown), and "Comments:" (empty text area). Below the form is a note about schema names and a note about default collation.
- Bottom:** Buttons for "Apply" and "Revert".

**Right Window (Table Creation):**

- Title Bar:** MySQL Workbench - Local instance 3306
- Toolbar:** Standard MySQL Workbench toolbar.
- Management Panel:** Shows Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, and Data Import/Restore.
- INSTANCE Panel:** Shows Startup / Shutdown, Server Logs, and Options File.
- PERFORMANCE Panel:** Shows Dashboard, Performance Reports, and Performance Schema Setup.
- SCHEMAS Panel:** Shows phpmyadmin and sys. The sys schema is selected.
- Central Area:** A dialog titled "Query 1" is open, showing the creation of a table named "new\_table" in the sys schema. The table definition includes a column named "Column Name" with a datatype of "String".
- Bottom:** Buttons for "Apply" and "Revert".

# Creazione

```
CREATE TABLE `gene` (  
    stable_id varchar(10)  
)
```

## Modifica

```
CREATE TABLE `gene` (
    stable_id VARCHAR(10)
)
```

- ALTER TABLE gene ADD COLUMN gene\_id int(10) unsigned NOT NULL;
- ALTER TABLE gene ADD COLUMN seq\_region\_start int(10) unsigned NOT NULL;
- ALTER TABLE gene ADD COLUMN seq\_region\_end int(10) unsigned NOT NULL;
- ALTER TABLE gene CHANGE stable\_id stable\_id varchar(128);

Lettura  
schema  
relazione

---

DESCRIBE gene;

# Eliminazione e Svuotamento

Eliminazione  
relazione

- `DROP TABLE table_name;`

Svuotamento  
tabella (si  
lavora sui  
record!)

- `TRUNCATE TABLE table_name;`
- `DELETE FROM table_name;`

# Creazione schema relazione completa

```
CREATE TABLE `gene` (
  `gene_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `biotype` varchar(40) NOT NULL,
  `analysis_id` smallint(5) unsigned NOT NULL,
  `seq_region_id` int(10) unsigned NOT NULL,
  `seq_region_start` int(10) unsigned NOT NULL,
  `seq_region_end` int(10) unsigned NOT NULL,
  `seq_region_strand` tinyint(2) NOT NULL,
  `display_xref_id` int(10) unsigned DEFAULT NULL,
  `source` varchar(40) NOT NULL,
  `description` text,
  `is_current` tinyint(1) NOT NULL DEFAULT '1',
  `canonical_transcript_id` int(10) unsigned NOT NULL,
  `stable_id` varchar(128) DEFAULT NULL,
  `version` smallint(5) unsigned DEFAULT NULL,
  `created_date` datetime DEFAULT NULL,
  `modified_date` datetime DEFAULT NULL);
```



Break

# Operazioni su record



Creazione



Lettura



Modifica



Cancellazione

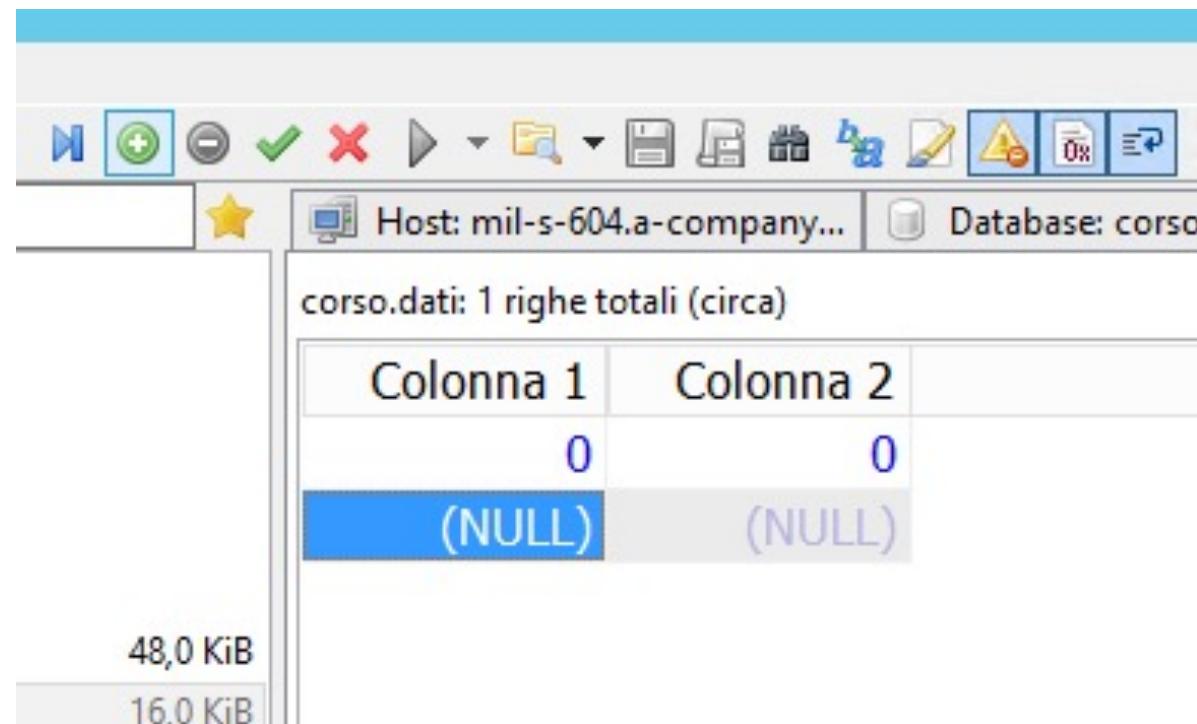
Creazione  
Record:  
popolare una  
tabella

`INSERT INTO table_name  
(campo1, campo2) VALUES  
(value1, value2);`

input da interfaccia grafica o da  
applicazione

Caricamento dati da file

# Creazione Record: popolare una tabella



# Creazione Record: popolare una tabella

- **INSERT INTO table\_name VALUES (value1, value2, value3,...);**
- **INSERT INTO gene (seq\_region\_start, seq\_region\_end, stable\_id) VALUES (7760, 7831, 'LCA\_19');**
- **INSERT INTO gene VALUES (554,'Mt\_tRNA',1,132907,577,647,1,6375890,'in sdc','mitochondrially encoded tRNA-Phe (UUU/C) [Source:HGNC Symbol;Acc:HGNC:7481]',1,1587,'ENSG00000210049',1,'2006-05-12 00:00:00','2006-05-12 00:00:00');**

# Esercizio

## Importazione dataset

# Creazione Tabella ed importazione

Accedo Remote Desktop

Accedo con terminale

- mysql -u user\_xxxx -h 127.0.0.1 -p
- use user\_xxxx;
- drop table gene;

mysql >

```
CREATE TABLE `gene` (  
  `gene_id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `biotype` varchar(40) NOT NULL,  
  `analysis_id` smallint(5) unsigned NOT NULL,  
  `seq_region_id` int(10) unsigned NOT NULL,  
  `seq_region_start` int(10) unsigned NOT NULL,  
  `seq_region_end` int(10) unsigned NOT NULL,  
  `seq_region_strand` tinyint(2) NOT NULL,  
  `display_xref_id` int(10) unsigned DEFAULT NULL,  
  `source` varchar(40) NOT NULL,  
  `description` text,  
  `is_current` tinyint(1) NOT NULL DEFAULT '1',  
  `canonical_transcript_id` int(10) unsigned NOT NULL,  
  `stable_id` varchar(128) DEFAULT NULL,  
  `version` smallint(5) unsigned DEFAULT NULL,  
  `created_date` datetime DEFAULT NULL,  
  `modified_date` datetime DEFAULT NULL)
```

mysql>

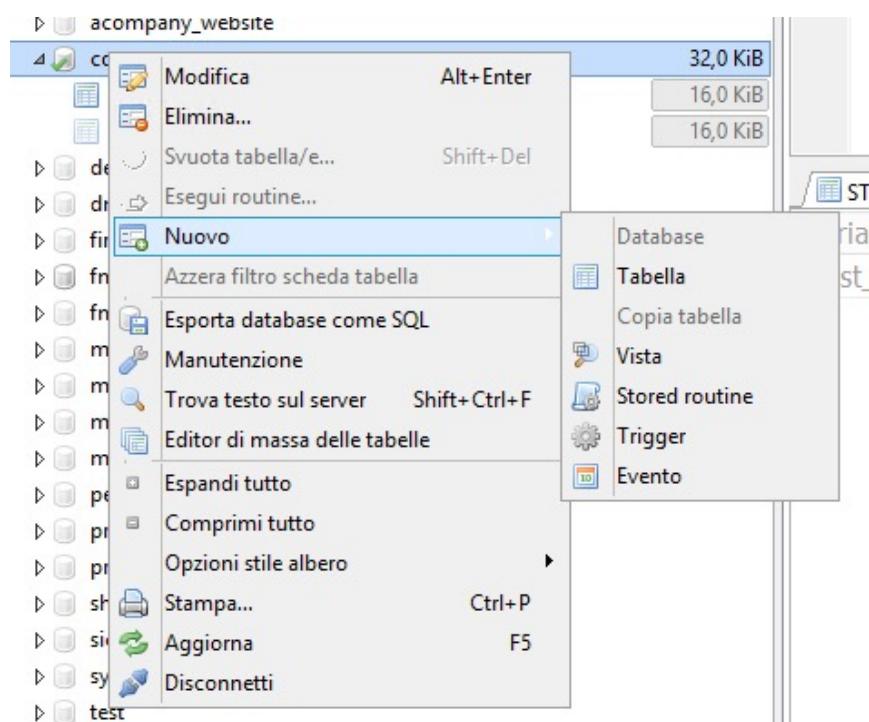
```
mysqlimport -u user_xxxx --fields-terminated-by='\t' --fields-escaped-by='\\' user_xxxx -L gene.txt
```

- SET GLOBAL local\_infile = true;
- SET GLOBAL binlog\_format = 'ROW';



Break

# Esercizio:Creazione Tabella

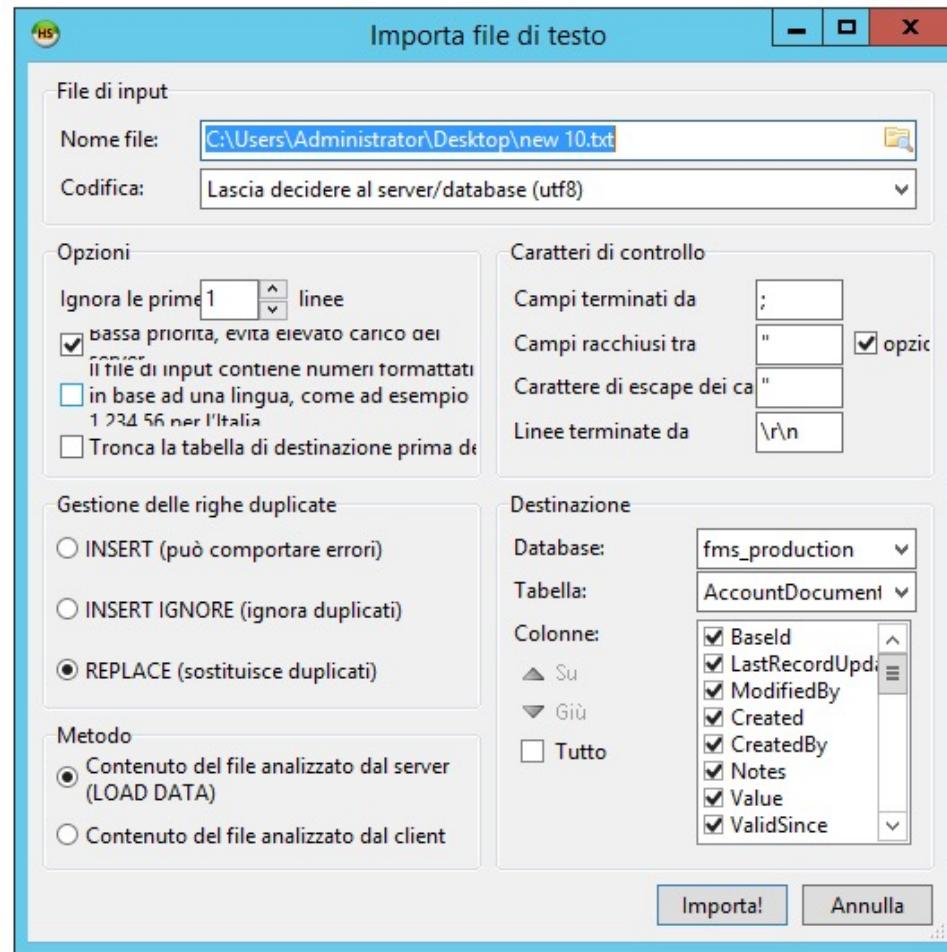


A screenshot of the MySQL Workbench 'Tabella' creation dialog. The 'Nome:' field contains 'Inserire nome tabella'. The 'Commento:' field is empty. The 'Colonne:' section shows one column named 'Colonna 1' with type 'INT'. There are tabs for Base, Opzioni, Indici, Chiavi esterne, Partizioni, and Codice CREATE. The 'Host' is mil-s-604.a-company... and the 'Database' is corso. The 'Tabella:' is Untitled.

#	Nome	Tipo di dati	Lunghezza/set	Senza s...	Permetti NULL
1	Colonna 1	INT			<input checked="" type="checkbox"/>

# Creazione Record: Importazione con Heidi Sql

```
find . -type f -name "*.txt" -exec rename 's/\.txt$/\.csv/' '{}' \;
```



# Creazione Record: importazione con CSV Kit



```
csvsql --dialect mysql --snifflimit 100000  
file_name.csv > sql_commands.sql
```



```
mysql -u [user] -p [password] [SCHEMA]  
< sql_commands.sql
```



```
LOAD DATA LOCAL INFILE '/tmp/gene.csv'  
INTO TABLE gene FIELDS TERMINATED BY  
'\t' ENCLOSED BY "" LINES TERMINATED BY  
'\n' IGNORE 1 ROWS;
```



Break

## Accedere ai dati

- SQL: structured query language
- Selezione di un set di record da una tabella
  - select \* from gene;
  - select \* from gene limit 10;
  - select \* from gene limit 10 order by sequence;
  - select stable\_id, seq\_region\_start, seq\_region\_end from gene where stable\_id='LRG\_89' order by stable\_id;
  - select stable\_id, seq\_region\_start, seq\_region\_end from gene where seq\_region\_start =5001 order by stable\_id;
  - select stable\_id from gene where created\_date<= '2004-12-01' order by created\_date limit 10;
  - select\* from gene where seq\_region\_end – seq\_region\_start < 10;

# ALIAS

Alias del nome di una tabella

```
select g.* from gene g where g.created_date  
> '2019-01-01' limit 10;
```

Alias di una colonna come output

```
select stable_id as gene_name from gene  
where created_date > '2019-01-01' limit  
10;
```

# Funzioni: introduzione

- Permettono di eseguire operazioni con i dati derivanti da una select per ottenere informazioni
- Esempio: conteggio delle righe di una query di selezione
  - `select count(*), source from gene group by source;`
  - `select count(*) as conteggio, source from gene where stable_id like 'LRG%' group by source;`
  - `select count(*), source from gene where stable_id like 'ENS%' group by source;`
  - `select count(*), YEAR(created_date) from gene where created_date < now() group by YEAR(created_date);`
  - `select count(*) as conteggio, year(created_date) as anno from gene where created_date < now() group by year(created_date) order by anno;`



# Funzioni

Eseguono operazioni sui valori  
selezionati in una query e  
generano un risultato

Si distinguono per tipo di dato

Funzioni per  
stringa (length,  
right, left, ucase)

Funzione per  
numeri ( max, min,  
round, count)

Funzioni per le  
date (now, datediff,  
month)

<a href="#"><u>INSERT()</u></a>	Insert substring at specified position up to specified number of characters
<a href="#"><u>INSTR()</u></a>	Return the index of the first occurrence of substring
<a href="#"><u>LCASE()</u></a>	Synonym for LOWER()
<a href="#"><u>LEFT()</u></a>	Return the leftmost number of characters as specified
<a href="#"><u>LENGTH()</u></a>	Return the length of a string in bytes
<a href="#"><u>LIKE</u></a>	Simple pattern matching
<a href="#"><u>LOAD_FILE()</u></a>	Load the named file
<a href="#"><u>LOCATE()</u></a>	Return the position of the first occurrence of substring
<a href="#"><u>LOWER()</u></a>	Return the argument in lowercase
<a href="#"><u>LPAD()</u></a>	Return the string argument, left-padded with the specified string
<a href="#"><u>LTRIM()</u></a>	Remove leading spaces

# String function

# Operatori

Source: MySql

## Aritmetici

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide
%	Modulo

## Di Confronto

Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to

## Logici

Operator	Description
ALL	TRUE if all of the subquery values meet the condition
AND	TRUE if all the conditions separated by AND is TRUE
ANY	TRUE if any of the subquery values meet the condition
BETWEEN	TRUE if the operand is within the range of comparisons
EXISTS	TRUE if the subquery returns one or more records
IN	TRUE if the operand is equal to one of a list of expressions
LIKE	TRUE if the operand matches a pattern
NOT	Displays a record if the condition(s) is NOT TRUE
OR	TRUE if any of the conditions separated by OR is TRUE
SOME	TRUE if any of the subquery values meet the condition

Individuare il record che ha il campo di testo con maggiore lunghezza.

## Esercizio

” ”

## Statement

```
select stable_id,  
length(stable_id)  
from gene where  
length(stable_id)=  
(select  
max(length(stable_id))  
from gene) limit 10 ;
```

Riepilogando

SELECT

FROM

WHERE

GROUP BY

ORDER BY

## Esercizio

Individuare quali campi  
hanno valori ridondanti

Contare il numero di  
ripetizioni per valore del  
campo con valori ridondanti.

Creazione  
record:  
popolare una  
tabella

---

**INSERT INTO table\_name VALUES  
(value1, value2, value3,...);**

---

Oppure popolare una tabella  
prendendo i dati da un'altra tabella

---

**INSERT INTO gene\_copy ( stable\_id  
 ) SELECT stable\_id FROM gene;**

---

**insert into gene\_copy select \* from  
gene;**

# Esercizio



Creare una tabella per uno dei campi con valori ridondanti trovato



Popolare la tabella con i soli valori unici



CREAZIONE



LETTURA

# Modifica Record

```
update gene set stable_id=ucase(stable_id);
```

```
update gene set created_date=now() where created_date>'2017-05-25' and created_date >'2016-03-28';
```

```
alter table dna add column length int unsigned not null;  
update gene set length=length(sequence);
```

```
alter table dna add column ripetizioni int unsigned not null;  
update dna set ripetizioni=(select (length(sequence) - length(replace(sequence,"GGG","")))/3 );
```

# delete

Create	create table dna (sequence_id int unsigned not null, sequence longtext not null);
Insert	insert into dna (select * from homo_sapiens_core_105_38.dna limit 1000);
Delete	delete from dna;
Truncate	truncate dna;
Delete	delete from dna where length(sequence) < 14;
Delete	delete from dna where (select (length(sequence) - length(replace(sequence,"GGG","")))/3 )=0;

- ```
SELECT * FROM gene WHERE source = 'insdc'
INTO OUTFILE '/var/lib/mysql-files/genes.csv'
FIELDS TERMINATED BY ',' ENCLOSED BY ""
LINES TERMINATED BY '\n';
```
- ```
mysqldump -u user_xxx -p -h 127.0.0.1
user_xxxx > backup.sql
```
- ```
mysql -h 127.0.0.1 -u user_xxx -p -e 'SELECT *
from ' > /tmp/myfile.txt
```

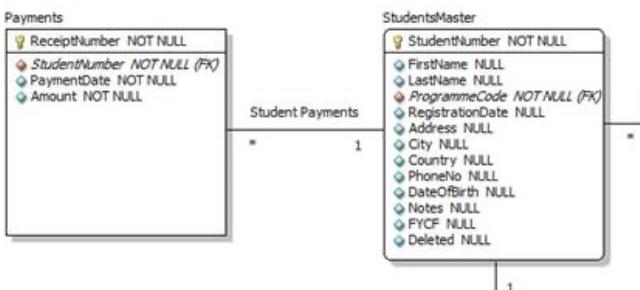
Export e  
Backup



Break

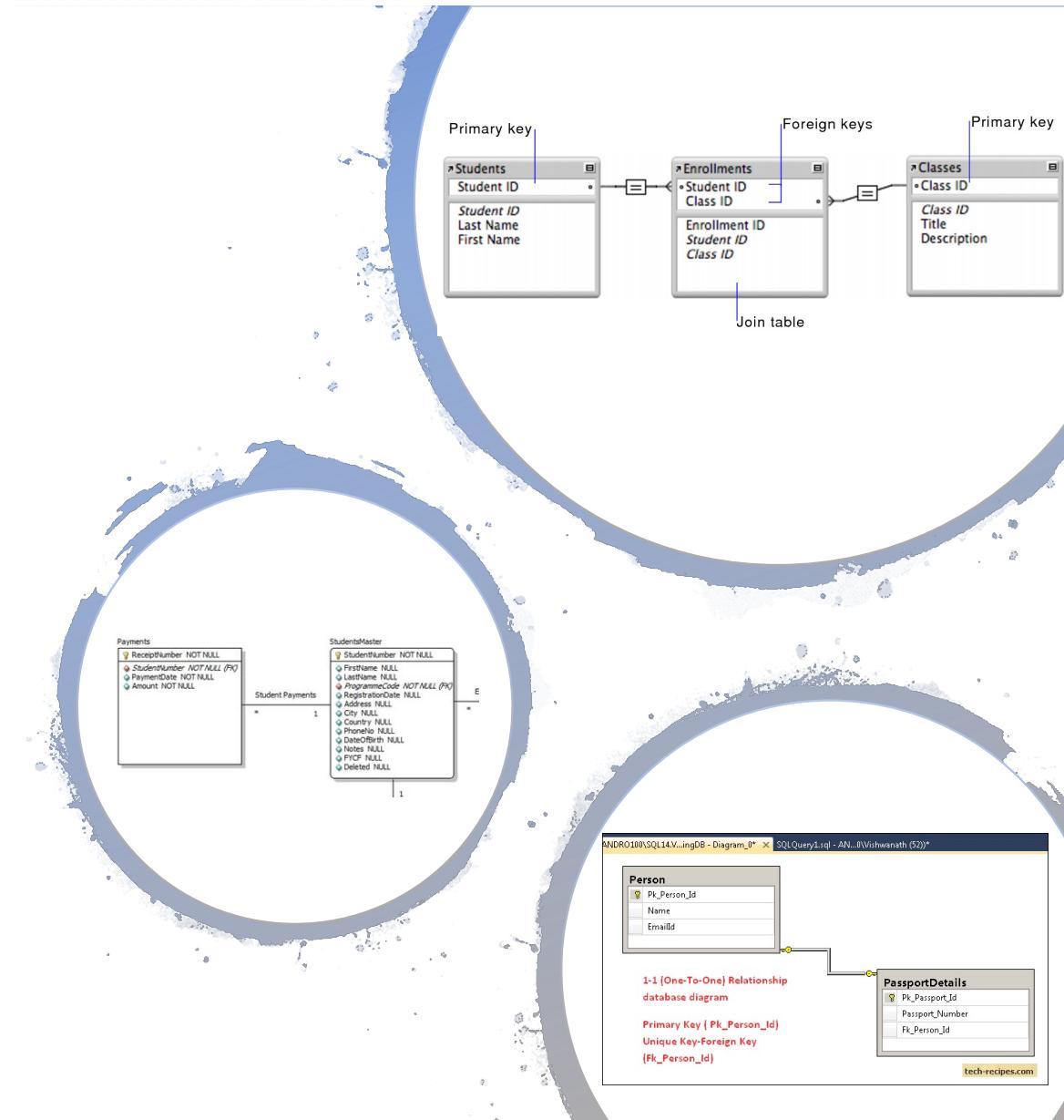
# Modello Entità Relazione

- In [informatica](#), nell'ambito della [progettazione dei database](#), il **modello entity-relationship** (detto anche **modello E-R**; in italiano: **modello entità-associazione** o con l'espressione desueta di '**modello entità-relazione**') è un modello teorico per la rappresentazione concettuale e grafica dei [dati](#) a un alto livello di [astrazione](#), formalizzato dal prof. Peter Chen nel 1976<sup>[1]</sup>. La vecchia dicitura '**entità-relazione**' è stata abbandonata in quanto, in passato, ha favorito la confusione tra la **relationship** del modello E-R (associazione concettuale) e la **relation** del [modello relazionale](#) (tabella logica).
- Source: [wikipedia](#)



# Tipo di associazione

- 1 ad 1
  - Passaporto - cittadino
- 1 ad N
  - Studente – pagamento
- N ad N
  - Studente - Classe



# Vincolo

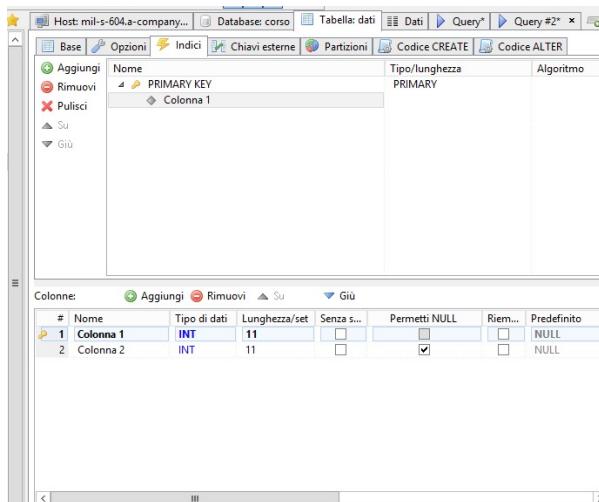
## Garanzia di integrità del dato.

### Interno

- Composta da una o più colonne
- Chiave primaria -> Primary key
  - Il record è unico rispetto alla chiave definita
  - NULL non ammessi
- Chiave unica e/o null -> Unique key
  - Il record è unico rispetto alla chiave definita
  - NULL ammessi

### Esterno

- Una chiave primaria è usata come riferimento in un record di un'altra tabella



# Chiave primaria

- Creiamo una chiave primaria nella tabella appena creata dai valori duplicati
  - Creazione della colonna
  - Definizione della chiave primaria
  - Attenzione alla convenzione di denominazione

```
CREATE TABLE `gene` (
`gene_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`biotype` varchar(40) NOT NULL,
`analysis_id` smallint(5) unsigned NOT NULL,
`seq_region_id` int(10) unsigned NOT NULL,
`seq_region_start` int(10) unsigned NOT NULL,
`seq_region_end` int(10) unsigned NOT NULL,
`seq_region_strand` tinyint(2) NOT NULL,
`display_xref_id` int(10) unsigned DEFAULT NULL,
`source` varchar(40) NOT NULL,
`description` text,
`is_current` tinyint(1) NOT NULL DEFAULT '1',
`canonical_transcript_id` int(10) unsigned NOT NULL,
`stable_id` varchar(128) DEFAULT NULL,
`version` smallint(5) unsigned DEFAULT NULL,
`created_date` datetime DEFAULT NULL,
`modified_date` datetime DEFAULT NULL,
PRIMARY KEY (`gene_id`))
```

# Chiave Esterna (Foreign Key)

---

creiamo nella tabella referente  
la colonna per la chiave esterna

---

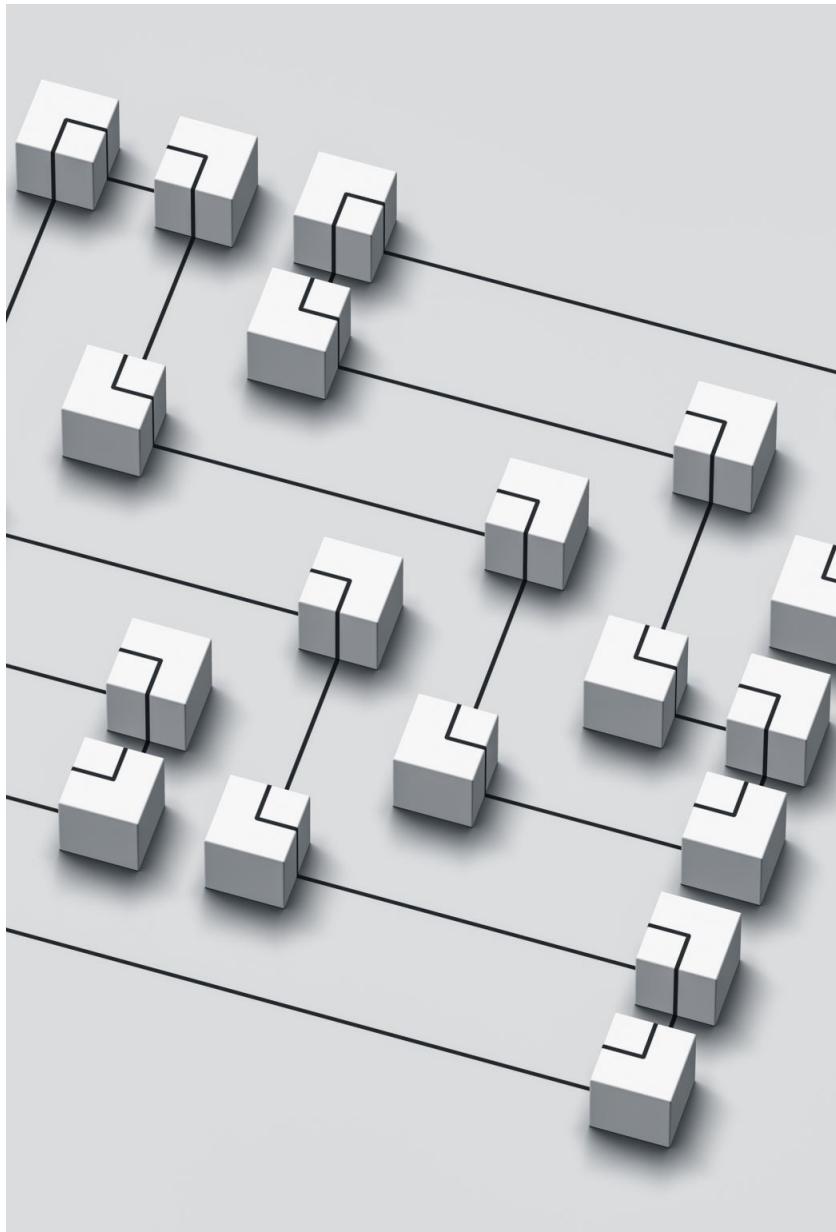
Attenzione alla convenzione di  
denominazione

---

Attenzione al tipo utilizzato

---

Leghiamo le due tabelle la  
referente e quella riferita



# Esercizio

---

- alter table gene add foreign key (seq\_region\_id) references gene(seq\_region\_id);
- alter table dna\_align\_feature add foreign key (seq\_region\_id) references dna(seq\_region\_id);
- alter table exon add foreign key (seq\_region\_id) references dna(seq\_region\_id);
- alter table assembly\_exception add foreign key (seq\_region\_id) references dna(seq\_region\_id);
- alter table density\_feature add foreign key (seq\_region\_id) references dna(seq\_region\_id);
- alter table ditag\_feature add foreign key (seq\_region\_id) references dna(seq\_region\_id);



## Cerchiamo l'errore

---

- Inviduiamo una chiave esterna mancante

# Chiave esterna sintassi



Chiave esterna tra tabella



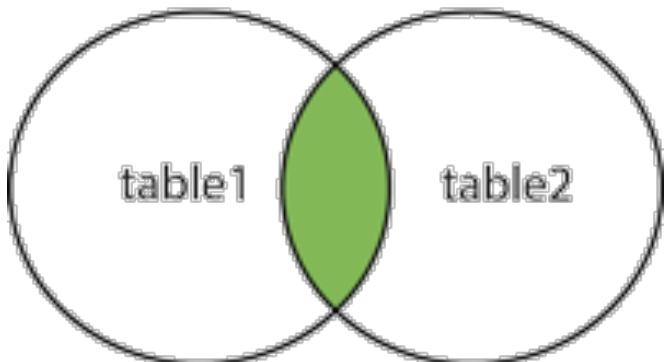
```
ALTER TABLE AthletePartecipation  
ADD CONSTRAINT IdSeason  
FOREIGN KEY (IdSeason)  
REFERENCES Season (Id);
```

---



Cosa manca ? Per  
completare la  
relazione!

## INNER JOIN

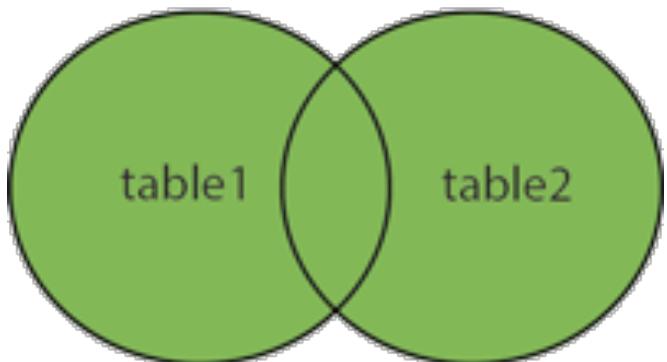


# SELECT INNER JOIN

```
SELECT column_name(s)  
FROM table1  
INNER JOIN table2  
ON table1.column_name = table2.column_name;
```



## FULL OUTER JOIN

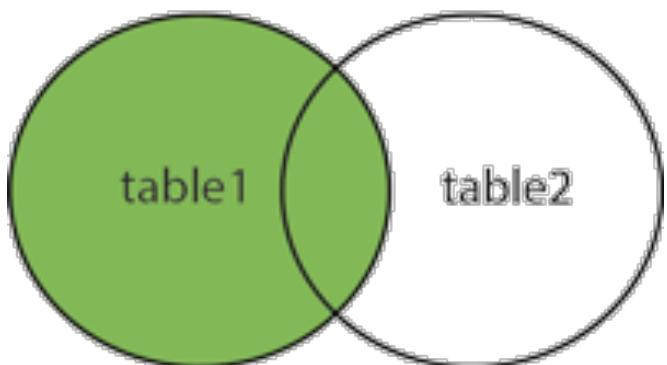


## FULL OUTER JOIN

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```



## LEFT JOIN

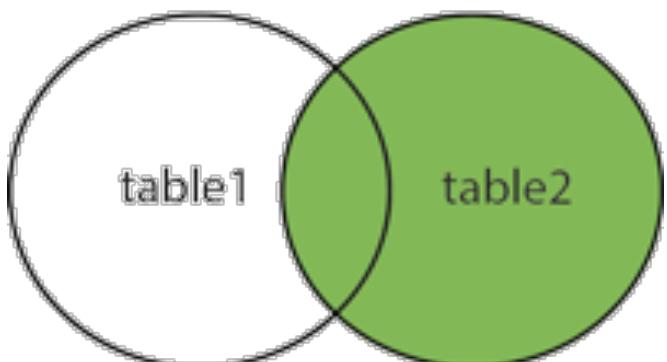


## LEFT JOIN

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```



## RIGHT JOIN



## RIGHT JOIN

```
SELECT column_name(s)  
FROM table1  
RIGHT JOIN table2  
ON table1.column_name = table2.column_na  
me;
```



# Ordine di esecuzione comandi SQL

```
SELECT DISTINCT <TOP_specification> <select_list>
FROM <left_table> <join_type>
JOIN <right_table> ON <join_condition>
WHERE <where_condition>
GROUP BY <group_by_list>
HAVING <having_condition>
ORDER BY <order_by_list>
```



# Indici

Albero di scansione della relazione.

```
ALTER TABLE `gene` ADD INDEX  
  `seq_region_idx`(`seq_region_id`, `seq_region_start`);  
ALTER TABLE `gene` ADD INDEX  
  `xref_id_index`(`display_xref_id`);  
ALTER TABLE `gene` ADD INDEX  
  `analysis_idx`(`analysis_id`);  
ALTER TABLE `gene` ADD INDEX  
  `stable_id_idx`(`stable_id`, `version`);  
ALTER TABLE `gene` ADD INDEX  
  `canonical_transcript_id_idx`  
  (`canonical_transcript_id`);
```



# Chiavi univoche

```
CREATE TABLE `gene_attrib` (
  `gene_id` int(10) unsigned NOT NULL DEFAULT '0',
  `attrib_type_id` smallint(5) unsigned NOT NULL DEFAULT '0',
  `value` text NOT NULL,
  UNIQUE KEY `gene_attribx` (`gene_id`,`attrib_type_id`,`value`(500)),
  KEY `type_val_idx` (`attrib_type_id`,`value`(40)),
  KEY `val_only_idx` (`value`(40)),
  KEY `gene_idx` (`gene_id`)
);
```

```
CREATE TABLE `gene_attrib` (
  `gene_id` int(10) unsigned NOT NULL DEFAULT '0',
  `attrib_type_id` smallint(5) unsigned NOT NULL DEFAULT '0',
  `value` text NOT NULL,
  KEY `type_val_idx` (`attrib_type_id`,`value`(40)),
  KEY `val_only_idx` (`value`(40)),
  KEY `gene_idx` (`gene_id`)
)
ALTER TABLE Persons ADD CONSTRAINT `gene_attribx` UNIQUE
(`gene_id`, `attrib_type_id`, `value`);
```

# SQL e R

```
install.packages("RMySQL")
```

```
library(RMySQL)
```

# Connessione

```
mydb =  
dbConnect(MySQL(),  
user='user',  
password='password',  
dbname='database_name  
, host='host')
```

# Istruzioni

```
dbListTables(mydb)
```

```
dbListFields(mydb, 'some_table')
```

```
dbSendQuery(mydb, 'drop table if exists some_table, some_other_table')
```

```
dbWriteTable(mydb, name='table_name', value=data.frame.name)
```

```
rs = dbSendQuery(mydb, "select * from some_table")
```

```
data = fetch(rs, n=-1)
```