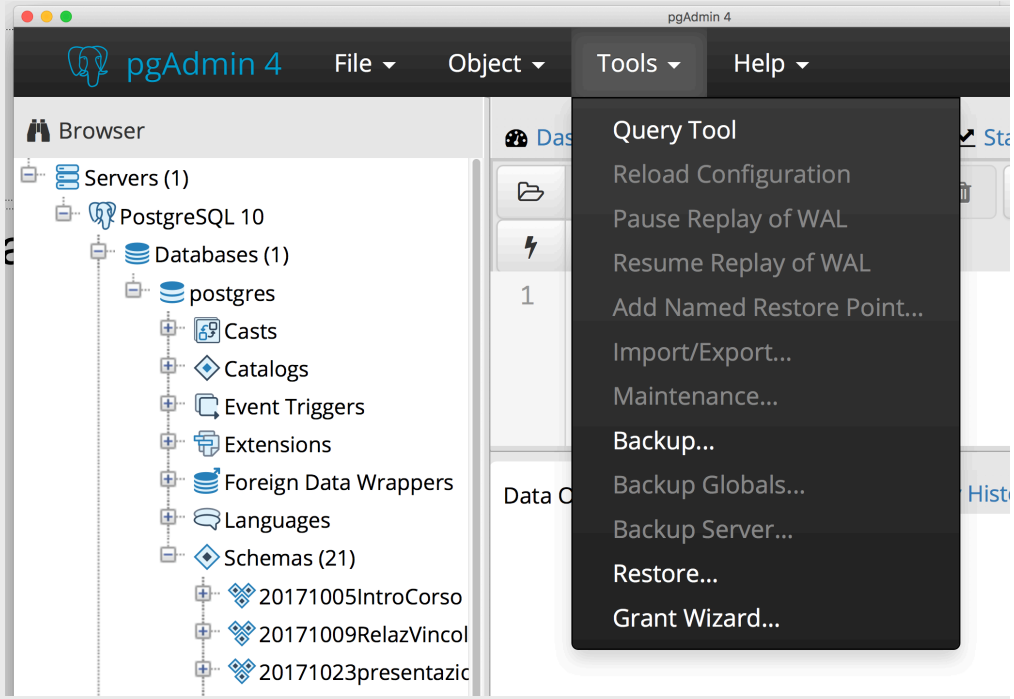
SQL

Esercitazioni pratiche

- Per SQL è possibile (e fondamentale) svolgere esercitazioni pratiche
- Verranno anche richieste come condizione per svolgere le prove parziali
- Soprattutto sono utilissime
- Si può utilizzare qualunque DBMS
  - IBM DB2, Microsoft SQL Server, Oracle, PostgresSQL o anche un servizio online (Sqliteonline)
- A lezione utilizziamo PostgresSQL e Sqliteonline

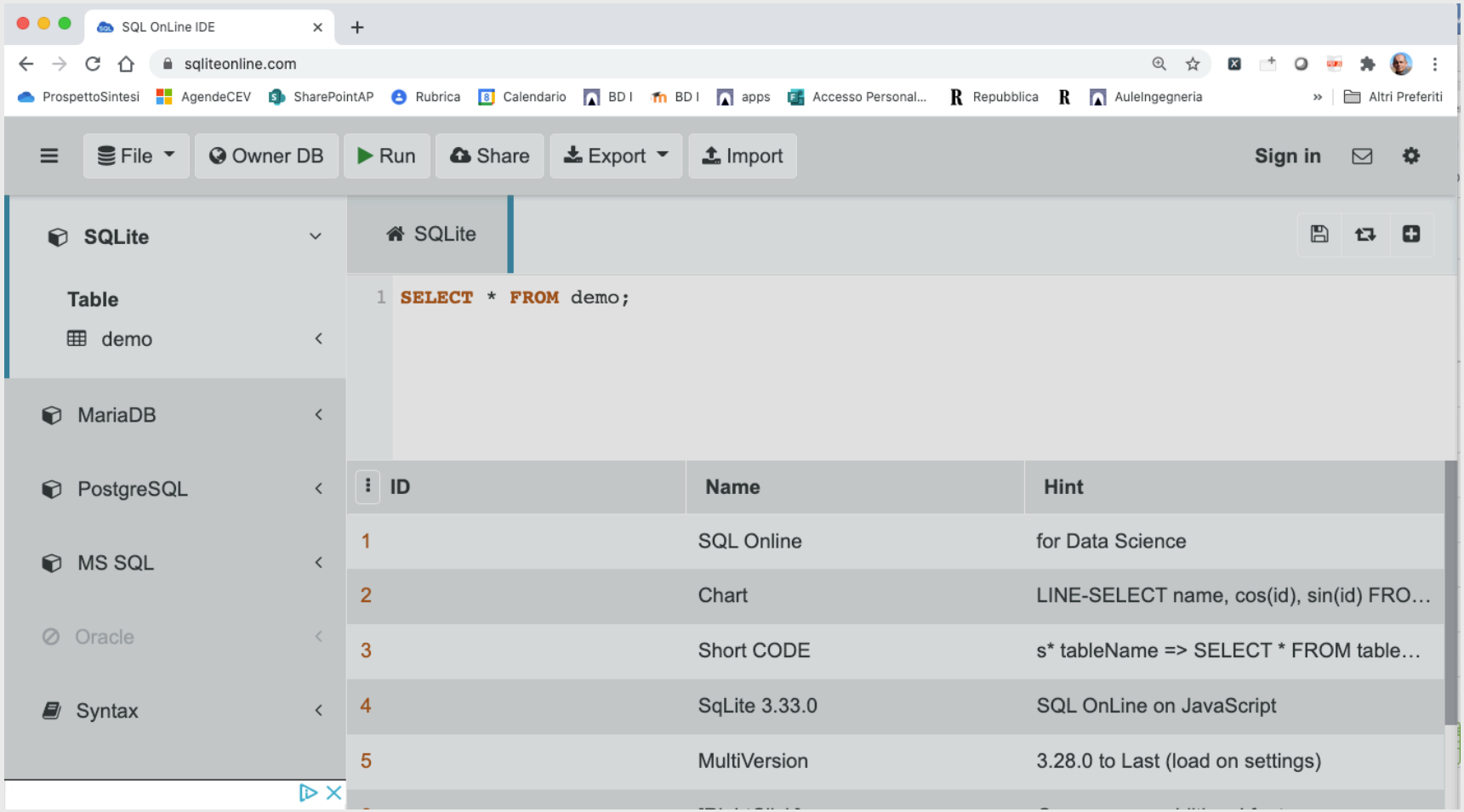
Come usare Postgres

- Scaricare
- Installare
- Lanciare pgAdmin
- Espandere l'albero a sinistra fino a "Schemas"
- Creare uno schema
- E poi
  - lavorare sugli elementi dello schema
  - oppure lanciare "Query tool" (SQL interattivo)



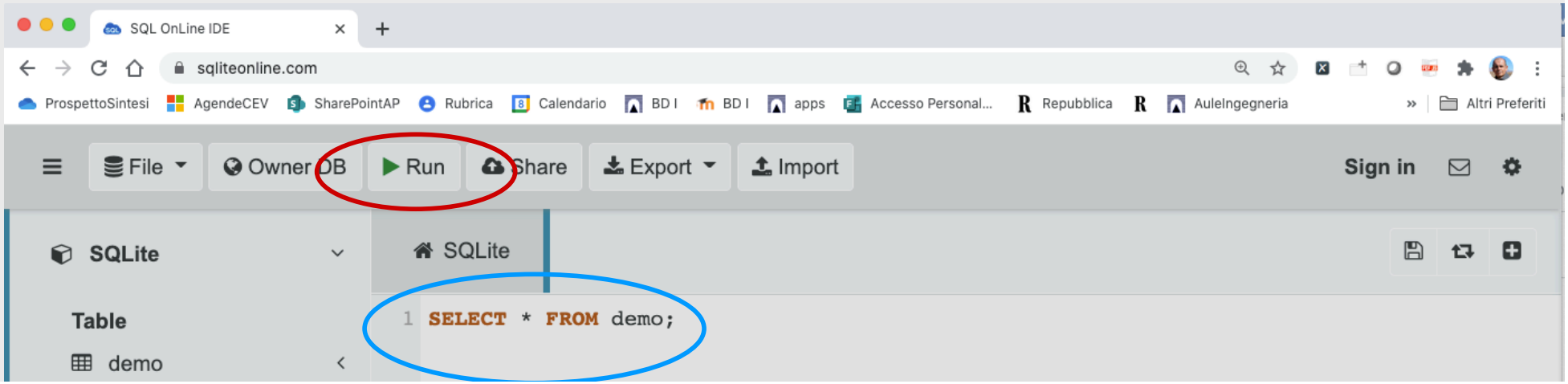
Come usare Sqliteonline

https://sqliteonline.com/



Come usare Sqliteonline

- I comandi vanno scritti nella finestra grande (al posto di
  ```
  SELECT * FROM demo;
  ```
- Vengono eseguiti premendo il pulsante Run
- Per inserire i dati si può caricare uno script (cioè un file di testo con una lista di comandi), come ad esempio (per I dati delle interrogazioni che vedremo fra poco):
  http://dia.uniroma3.it/~atzeni/didattica/BDN/20202021/protected/BD-04-esempiSQL2020%20persone.sql



CREATE TABLE, esempi

```
CREATE TABLE corsi(
    codice numeric NOT NULL PRIMARY KEY,
    titolo character(20) NOT NULL,
    cfu numeric NOT NULL)

CREATE TABLE esami(
    corso numeric REFERENCES corsi (codice),
    studente numeric REFERENCES studenti,
    data date NOT NULL,
   voto numeric NOT NULL,
   PRIMARY KEY (corso, studente))
```

La chiave primaria viene definita come NOT NULL anche se non lo specifichiamo (in Postgres)

DDL, in pratica

- In molti sistemi si utilizzano strumenti diversi dal codice SQL per definire lo schema della base di dati

SQL, operazioni sui dati

- interrogazione:
  - SELECT
- modifica:
  - INSERT, DELETE, UPDATE

Inserimento

(necessario per gli esercizi)

INSERT INTO Tabella [(Attributi)]
VALUES(Valori)

oppure

INSERT INTO Tabella [(Attributi)]
SELECT ...
(vedremo più avanti)

```
INSERT INTO Persone VALUES ('Mario',25,52)

INSERT INTO Persone(Nome, Reddito, Eta)
      VALUES('Pino',52,23)

INSERT INTO Persone(Nome, Reddito)
      VALUES('Lino',55)
```

Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Paternità

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Persone

Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Esercizi

- Definire la base di dati per gli esercizi
  - installare un sistema
  - creare lo schema
  - creare le relazioni (CREATE TABLE)
  - inserire i dati

  - eseguire le interrogazioni
    - suggerimento, per chi usa Postgres: usare schemi diversi

      set search_path to <*nome schema*>

```
create table persone (
  nome char (10) not null primary key,
  eta numeric not null,
  reddito numeric not null);
create table paternita (
  padre char (10) references persone,
  figlio char (10) primary key references persone);
...
insert into Persone values('Andrea',27,21);
...
insert into Paternita values('Sergio','Franco');
...
```

Vedere file:

http://dia.uniroma3.it/~atzeni/didattica/BDN/20202021/protected/BD-04-esempiSQL2020%20persone.sql

Vediamo gli esempi

- con RelaX

  http://dbis-uibk.github.io/relax/calc/gist/1362a9bb84dd2e4052561b714613b1de

- con Sqliteonline

  https://sqliteonline.com/

  con i dati

  http://dia.uniroma3.it/~atzeni/didattica/BDN/20202021/protected/BD-04-esempiSQL2020%20persone.sql

Istruzione SELECT (versione base)

SELECT ListaAttributi
FROM ListaTabelle
[WHERE Condizione]

- "target list"
- clausola FROM
- clausola WHERE

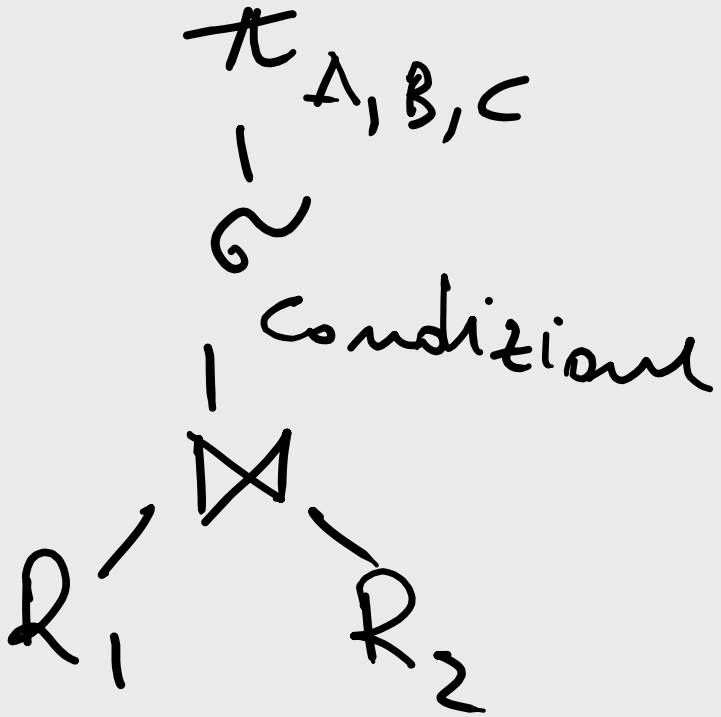
Intuitivamente

SELECT ListaAttributi
FROM ListaTabelle
[WHERE Condizione]

- Prodotto cartesiano di ListaTabelle
- Selezione su Condizione
- Proiezione su ListaAttributi

Intuitivamente

SELECT A, B, C
FROM R1, R2
[WHERE Condizione]



Una sola relazione

- Selezioni e proiezioni

Selezione e proiezione

- Nome e reddito delle persone con meno di trenta anni

$$PROJ_{Nome,\ Reddito}(SEL_{Eta<30}(Persone))$$

```
select nome, reddito
from persone
where eta < 30
```

Selezione, senza proiezione

- Nome, età e reddito delle persone con meno di trenta anni

$$SEL_{Eta<30}(Persone)$$

```
select *
from persone
where eta < 30
```

Proiezione, senza selezione

- Nome e reddito di tutte le persone

$$PROJ_{Nome,\ Reddito}(Persone)$$

```
select nome, reddito
from persone
```

Proiezione, con ridenominazione

- Nome e reddito di tutte le persone

$$REN_{Anni \leftarrow Eta}(PROJ_{Nome,\ Eta}(Persone))$$

```
select nome, eta as anni
from persone
```

Proiezione, attenzione

```
select
    madre
from maternita
```

```
select distinct
    madre
from maternita
```

Condizione complessa

```
select *
from persone
where reddito > 25
    and (eta < 30 or eta > 60)
```

Selezione, proiezione e join

- I padri di persone che guadagnano più di 20

$$\text{PROJ}_{\text{Padre}}(\text{paternita}\ \text{JOIN}_{\text{Figlio}=\text{Nome}}\ \text{SEL}_{\text{Reddito}>20}(\text{persone}))$$

```
select distinct padre
from persone, paternita
where figlio = nome and reddito > 20
```

- Le persone che guadagnano più dei rispettivi padri; mostrare nome, reddito e reddito del padre

$$\text{PROJ}_{\text{Nome, Reddito, RP}}\,(\text{SEL}_{\text{Reddito}>\text{RP}}\,(\text{REN}_{\text{NP,EP,RP}\leftarrow\text{Nome,Eta,Reddito}}(\text{persone})\ \text{JOIN}_{\text{NP=Padre}}\ (\text{paternita JOIN}_{\text{Figlio =Nome}}\ \text{persone})))$$

```
select f.nome, f.reddito, p.reddito
from persone p, paternita, persone f
where p.nome = padre and
   figlio = f.nome and
   f.reddito > p.reddito
```

SELECT, con ridenominazione del risultato

```
select figlio, f.reddito as reddito,
    p.reddito as redditoPadre
from persone p, paternita, persone f
where p.nome = padre and figlio = f.nome
      and f.reddito > p.reddito
```