DOMANDE:

- Cosa si intende per API REST ?
- Cosa si intende per servizio SOAP ?
- Cos'è un database relazionale ?
- Cos'è un database NoSQL ?
- Cos'è un ORM ? Fai almeno un esempio.
- Cos'è la SQL Injection ?
- Cos'è l'autenticazione a 2 fattori? Descrivi brevemente un esempio.
- Descrivi brevemente un metodo sicuro per salvare le password degli utenti sul DB.
- Cos'è una Sticky Session in un sistema con Load Balancing?

ESERCIZIO:

SPIEGAZIONE DATABASE:

Il file "**excercise01.sqlite**" contiene un DB **SQLite** contenente diverse tabelle.
La tabella principale è "**records**", la quale contiene numerosi record estratti da un censimento americano.

Ecco i campi della tabella:

- **id**: a unique id number for each record
- **age**: a continuous variable representing an individual's age
- **workclass_id**: foreign key to the workclasses table, representing the broad class of occupation of an individual
- **education_level_id**: foreign key to the education_levels table, representing the highest level of education an individual received
- **education_num**: a continuous variable representing an individual's current education level
- **marital_status_id**: foreign key to the marital_statuses table, representing an individual's marital status
- **occupation_id**: foreign key to the occupations table, representing an individual's occupation
- **race_id**: foreign key to the races table, representing an individual's race
- **sex_id**: foreign key to the sexes table, representing an individual's sex

- **capital_gain**: a continuous variable representing post-social insurance income, in the form of capital gains.
- **capital_loss**: a continuous variable representing post-social insurance losses, in the form of capital losses.
- **hours_week**: a continuous variable representing the number of hours per week an individual worked.
- **country_id**: foreign key to the countries table, representing an individual's native country
- **over_50k**: a boolean variable representing whether the individual makes over $50,000/year. A value of 1 means that the person makes greater than $50,000/year and a value of 0 means that the person makes less than or equal to $50,000/year.

Tutti i campi che finiscono con "**_id**" fanno riferimento ad altre tabella contenute nel database.

ESERCIZI:

1) Scrivi una query che estragga il **numero di persone** con **meno di 30 anni** che guadagnano **più di 50.000 dollari l'anno**

2) Scrivi una query che riporti il **guadagno di capitale medio** per ogni categoria lavorativa

3) Scegli la tecnologia che preferisci tra **.NET Core**, **Node.js**, **PHP, Java** e **APEX**, realizza quindi un **servizio web** che legga i dati dall'**SQLite** ed esponga le seguenti **Web API** in **JSON**:

    - **Lista dei record denormalizzati**, cioè ogni entità deve contenere anche tutte le informazioni derivanti dalle tabelle secondarie del DB.
      L'API deve essere realizzata in **GET** e avere una **paginazione parametrica** (cioè tramite l'url è possibile definire offset e count)

    - **Statistiche** aggregate filtrate in base ad alcuni parametri significativi.
      L'API, realizzabile in **GET** o in **POST**, deve soddisfare questa documentazione fornita dal cliente:

INPUT: • aggregationType: "age", "education_level_id", "occupation_id • aggregationValue: int
OUTPUT: {

```
        "aggregationType": "string",
        "aggregationFilter": int,
        "capital_gain_sum": float,
        "capital_gain_avg": float,
        "capital_loss_sum": float,
        "capital_loss_avg": float,
        "over_50k_count": int,
        "under_50k_count": int
}
```

ESEMPIO:

Passando i parametri "aggregationType" = "age" e "aggregationValue" = 30 si ottengono le statistiche per tutti coloro che hanno 30 anni

4) Esponi inoltre, tramite il medesimo servizio web, un endpoint che faccia il download in formato **CSV** di tutti i dati **denormalizzati**
   (cioè ogni riga deve contenere sia il record che tutti i dati relazionati dalle altre tabelle)

PER LA CONSEGNA:

- Repository **GIT** su GitHub (o qualsiasi sistema alternativo) dal quale poter scaricare l'intero progetto e visionare lo storico dei commit