Basi di Dati - Informatica umanistica Esercitazione (Lab)

a.a. 2017 - 2018

Vinícius Monteiro de Lira
(vinicius.monteirodelira@isti.cnr.it)

Esercitazione

Mini-project!

openflights.org

Esercitazione (Summary)

Progettazione ER

DDL (Data Definition Language):

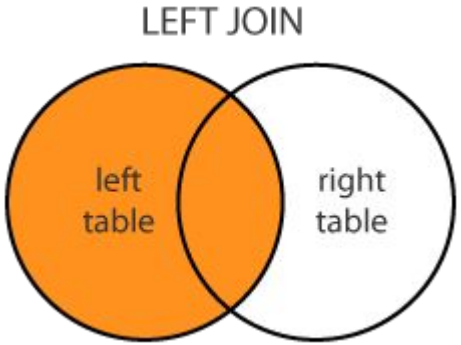
- *create*, *alter*, *add*, *drop;*
  Permette di creare, modificare o eliminare gli oggetti in un database.

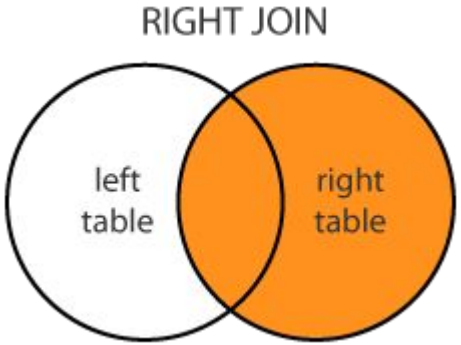
DML (Data Manipulation Language):

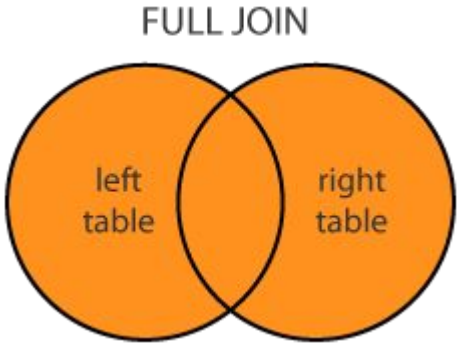
- *SELECT* per la ricerca;
  Usando inner, *left*, *right*, *full*, *exists*, *group by*, *order by*, *limit*

Ricordando...

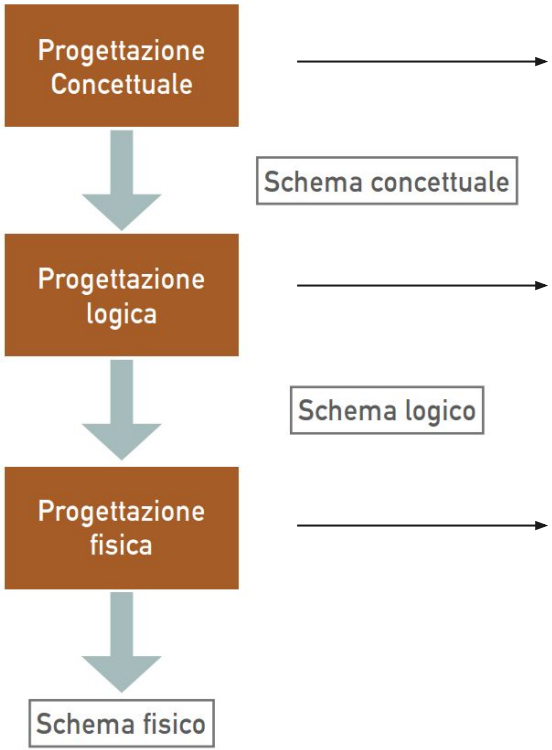
OUTER JOINS







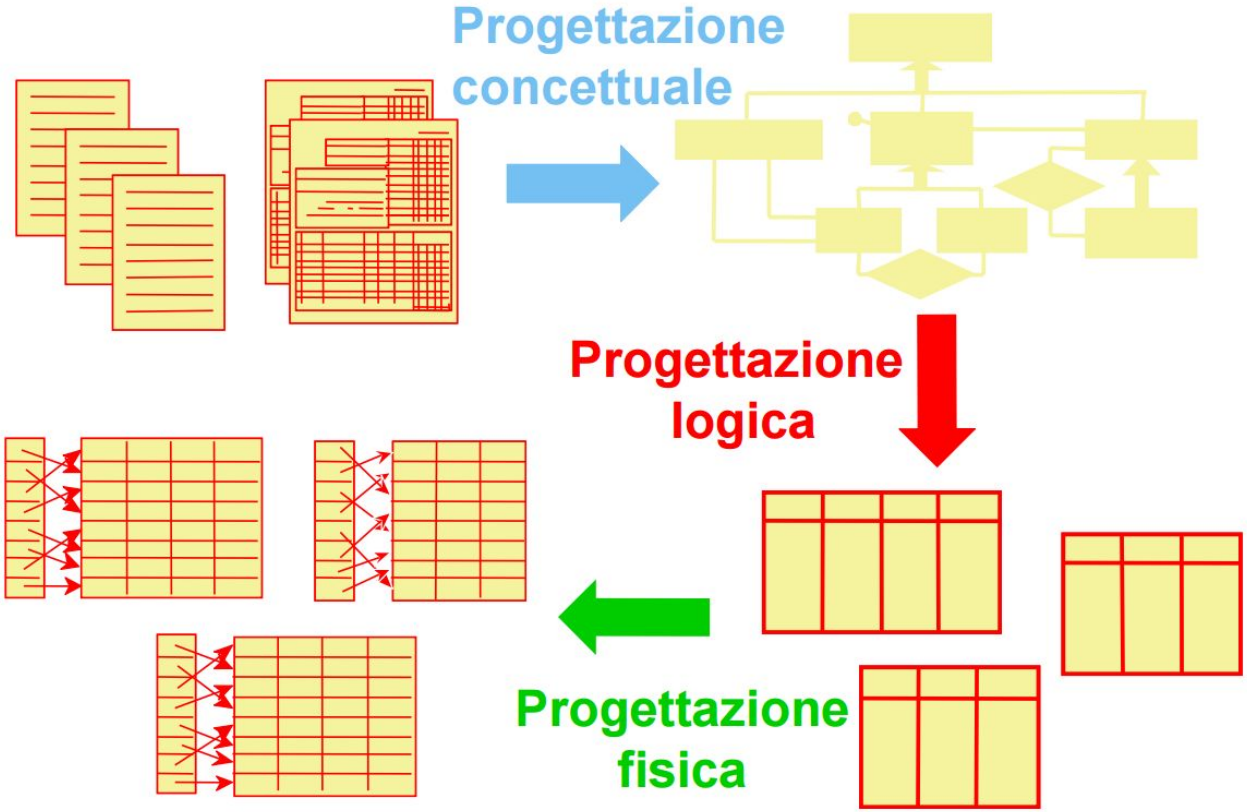
Ricordando...



Progettazione Concettuale. Dare una rappresentazione formale della realtà di interesse.

Progettazione Logica. Tradurre lo schema concettuale nelle strutture dati della base di dati.

Progettazione Concettuale. Completare lo schema logico con la specifica dei parametri fisici di memorizzazione dei dati.


Progettazione
concettuale
Progettazione
logica
Progettazione
fisica

Modello Entità-Relazione (E-R)

È un modello concettuale dei dati:

➤ descrive in modo formale (utilizzando opportuni costrutti) con un elevato livello di astrazione la realtà di interesse.

Costrutti:
➤ Entità
➤ Relazioni (e le corrispondenti cardinalità)
➤ Attributi (e le corrispondenti cardinalità)
➤ Identificatori delle entità
➤ Generalizzazioni

Let's work

https://openflights.org/

Routes		Airports		Airlines		Planes	
MUC ↔ DXB	6	Munich International (MUC)	39	Unknown	29	Airbus A320	10
HAM ↔ MUC	3	London Heathrow Airp (LHR)	17	Delta Air Lines	12	Airbus A320-200	6
SIN ↔ DPS	3	Dubai International (DXB)	14	Lufthansa	11	Airbus 320	3
SFO ↔ ATL	3	Frankfurt am Main In (FRA)	13	British Airways	11	Airbus A330-300	3
MUC ↔ AUH	2	Amsterdam Airport Sc (AMS)	13	Emirates	11	Boeing 737-800	3
MUC ↔ BCN	2	John F Kennedy Inter (JFK)	12	American Airlines	7	Airbus A319	3
DXB ↔ CMB	2	Singapore Changi Air (SIN)	11	Aeroflot Russian Airlines	7	Embraer 190	2
FRA ↔ LHR	2	Las Américas Interna (SDQ)	11	KLM Royal Dutch Airlines	6	A340-600	2
BUD ↔ FCO	2	Copenhagen Kastrup A (CPH)	8	Singapore Airlines	6	Airbus A321-200	2
LHR ↔ JFK	2	Hamburg Airport (HAM)	7	LAN Airlines	6	Airbus A319-100	2

Tasks:

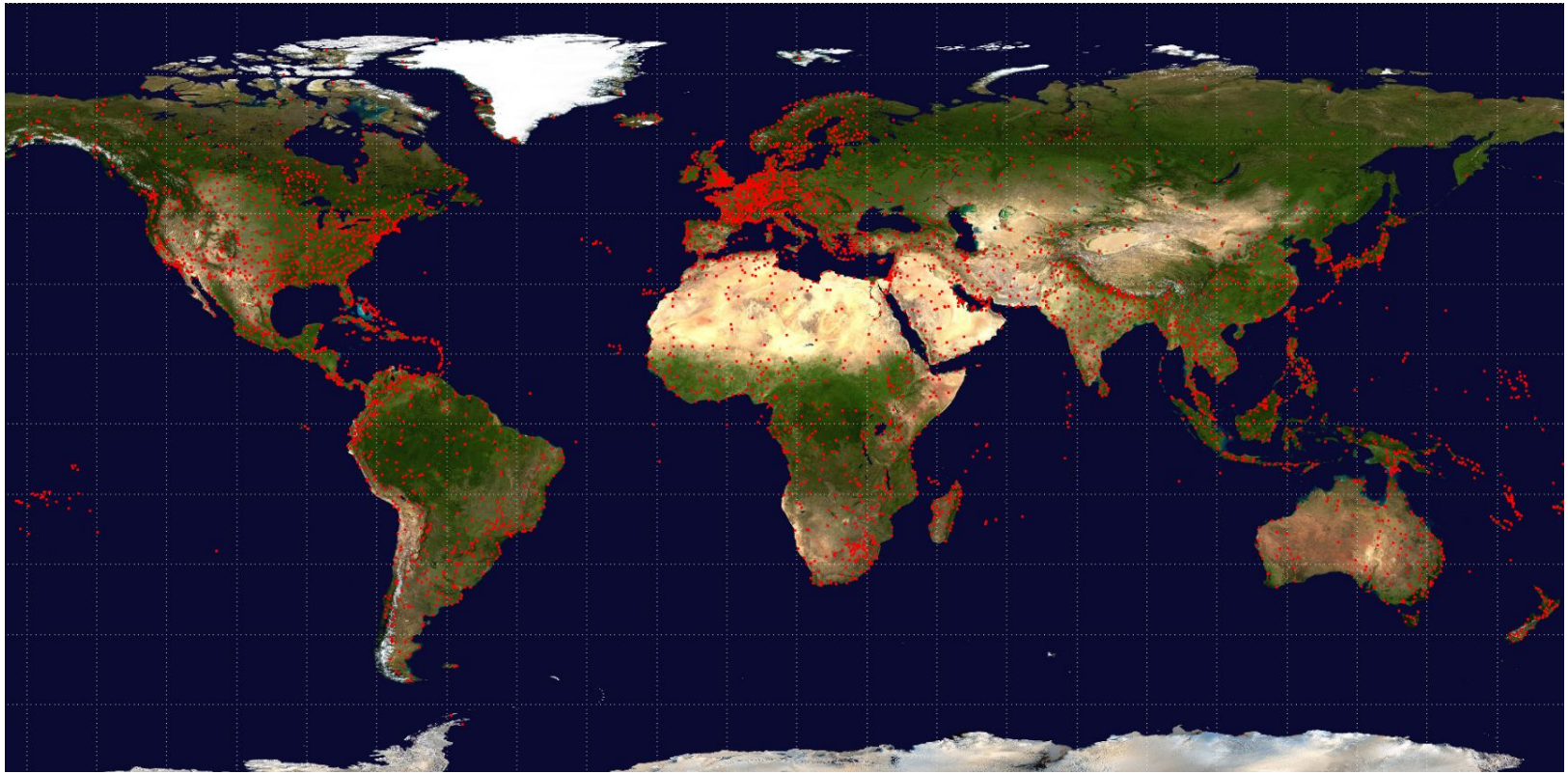
1. Creazione del Modello Entità-Relazione (E-R)
2. Creazione del Modello Logico
3. Creazione del database usando SQL Lite
4. Importazione dei dati
5. Creazione di reports (queries)

Task 1. Modello Entità-Relazione (E-R)

Requisiti (Airport)

Airport ID	Unique OpenFlights identifier for this airport.
Name	Name of airport. May or may not contain the **City** name.
City	Main city served by airport. May be spelled differently from **Name**.
Country	Country or territory where airport is located. See countries.dat to cross-reference to ISO 3166-1 codes.
IATA	3-letter IATA code. Null if not assigned/unknown.
ICAO	4-letter ICAO code. Null if not assigned.
Latitude	Decimal degrees, usually to six significant digits. Negative is South, positive is North.
Longitude	Decimal degrees, usually to six significant digits. Negative is West, positive is East.
Altitude	In feet.
Timezone	Hours offset from UTC. Fractional hours are expressed as decimals, eg. India is 5.5.
DST	Daylight savings time. One of E (Europe), A (US/Canada), S (South America), O (Australia), Z (New Zealand), N (None) or U (Unknown). *See also: Help: Time*
Tz database time zone	Timezone in "tz" (Olson) format, eg. "America/Los_Angeles".
Type	Type of the airport. Value "airport" for air terminals, "station" for train stations, "port" for ferry terminals and "unknown" if not known. *In airports.csv, only type=airport is included.*
Source	Source of this data. "OurAirports" for data sourced from OurAirports, "Legacy" for old data not matched to OurAirports (mostly DAFIF), "User" for unverified user contributions. *In airports.csv, only source=OurAirports is included.*

Requisiti (Airport)



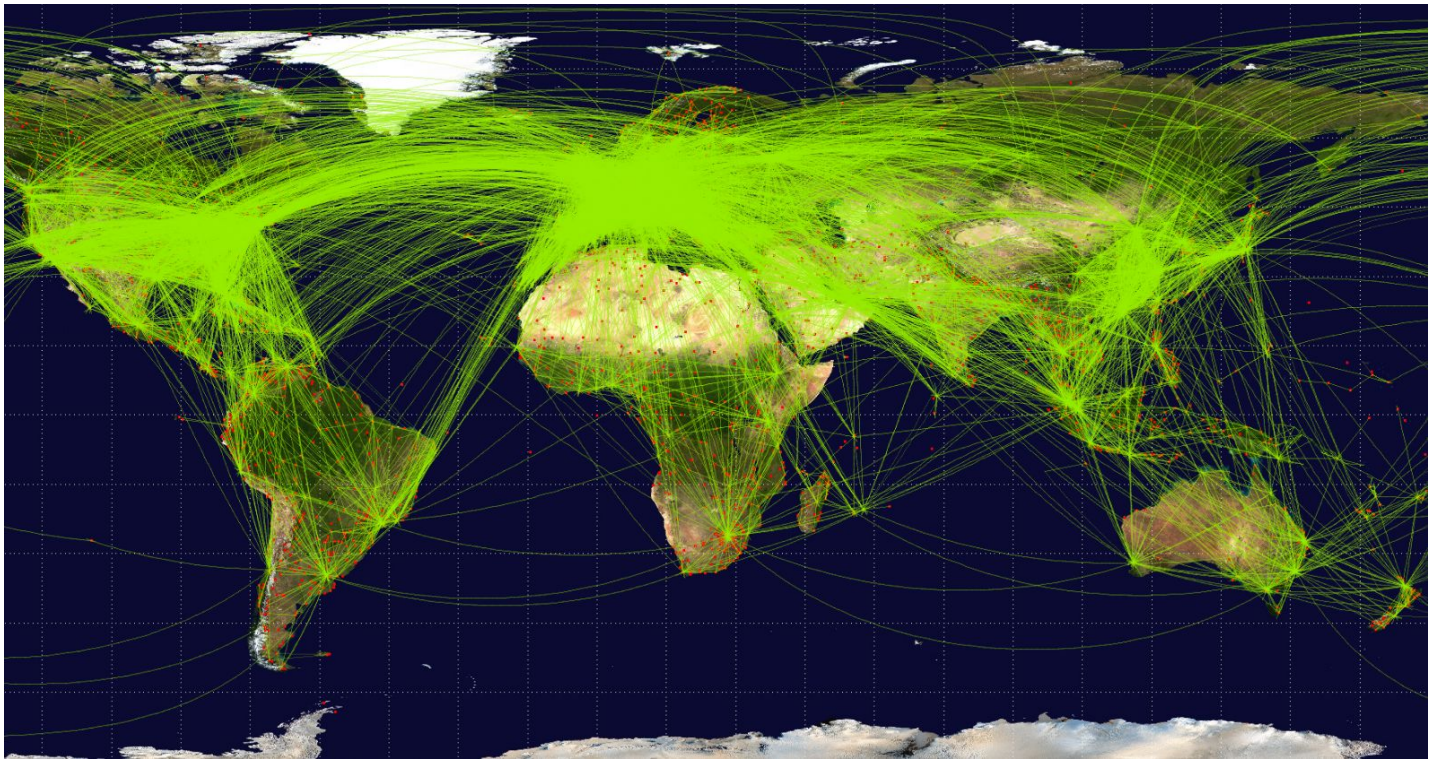
Requisiti (Airlines)

Airline ID	Unique OpenFlights identifier for this airline.
Name	Name of the airline.
Alias	Alias of the airline. For example, All Nippon Airways is commonly known as "ANA".
IATA	2-letter IATA code, if available.
ICAO	3-letter ICAO code, if available.
Callsign	Airline callsign.
Country	Country or territory where airline is incorporated.
Active	"Y" if the airline is or has until recently been operational, "N" if it is defunct. This field is *not* reliable: in particular, major airlines that stopped flying long ago, but have not had their IATA code reassigned (eg. Ansett/AN), will incorrectly show as "Y".

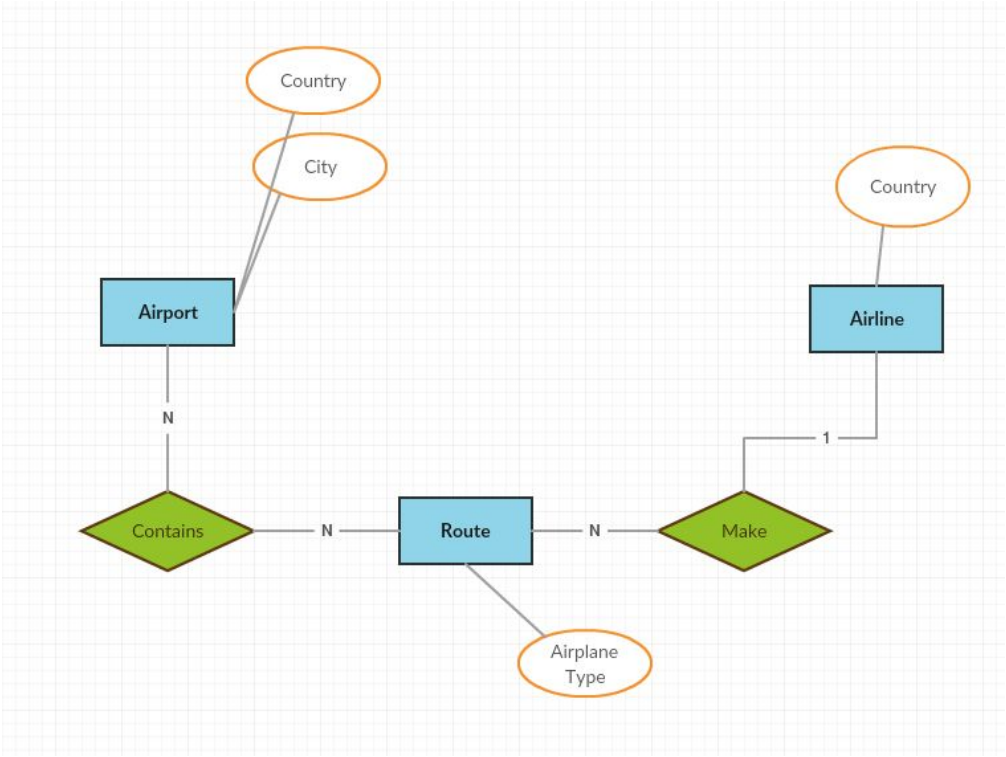
Requisiti (Route)

Airline	2-letter (IATA) or 3-letter (ICAO) code of the airline.
Airline ID	Unique OpenFlights identifier for airline (see Airline).
Source airport	3-letter (IATA) or 4-letter (ICAO) code of the source airport.
Source airport ID	Unique OpenFlights identifier for source airport (see Airport)
Destination airport	3-letter (IATA) or 4-letter (ICAO) code of the destination airport.
Destination airport ID	Unique OpenFlights identifier for destination airport (see Airport)
Codeshare	"Y" if this flight is a codeshare (that is, not operated by *Airline*, but another carrier), empty otherwise.
Stops	Number of stops on this flight ("0" for direct)
Equipment	3-letter codes for plane type(s) generally used on this flight, separated by spaces

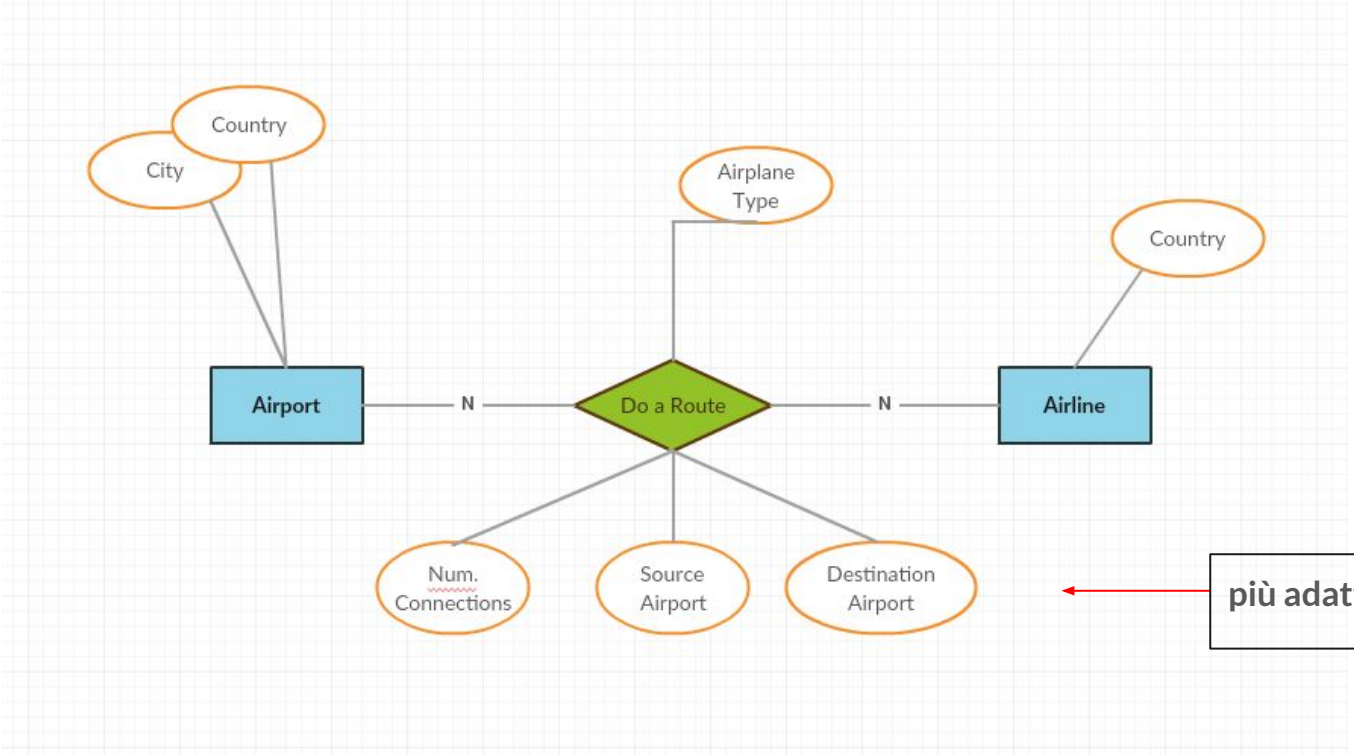
Requisiti (Route)



Task 1. Modello Entità-Relazione (E-R)



Task 1. Modello Entità-Relazione (E-R)



Task 2. Creazione del Modello Logico

```
CREATE TABLE airport (
        airportID INTEGER PRIMARY KEY UNIQUE NOT NULL,
        name VARCHAR (300) NOT NULL,
        city VARCHAR (300) NOT NULL,
        country VARCHAR (300) NOT NULL,
        codeIATA CHAR (3),
        codeICAO CHAR (4),
        latitude DECIMAL CONSTRAINT constraint_valid_latitude CHECK (latitude BETWEEN -90 AND 90),
        longitude DECIMAL CONSTRAINT constraint_valid_longitude CHECK (longitude BETWEEN -180 AND 180),
        altitude INTEGER,
        timezone INTEGER,
        dst CHAR (1) CONSTRAINT constraint_valid_dst CHECK (dst == 'E' OR dst == 'A' OR dst == 'S' OR
        dst == 'O' OR dst == 'Z' OR dst == 'N' OR dst == 'U' OR dst IS NULL),
        tz VARCHAR (100),
        type VARCHAR (100),
        source VARCHAR (400)
);
```

Task 2. Creazione del Modello Logico

```
CREATE TABLE airline (
        airlineID INTEGER PRIMARY KEY UNIQUE,
        name VARCHAR (300),
        alias VARCHAR (50),
        codeIATA CHAR (2),
        codeICAO CHAR (3),
        callSign VARCHAR (100),
        country VARCHAR (50),
        active CHAR (1)
);
```

Task 2. Creazione del Modello Logico

```
CREATE TABLE route (
        airline CHAR (3),
        airlineID INTEGER CONSTRAINT fk_airline REFERENCES airline (airlineID) ON DELETE CASCADE,
        sourceAirport CHAR (4),
        sourceAirportID INTEGER CONSTRAINT fk_sourceAirport REFERENCES airport (airportID) ON DELETE
        SET NULL,
        destinationAirport CHAR (4),
        destinationAirportID INTEGER CONSTRAINT fk_destinationAirport REFERENCES airport (airportID) ON
        DELETE SET NULL,
        codeShare CHAR (1),
        stops INTEGER,
        planetype CHAR (3)
);
```

Task 3. Creazione del database usando SQL Lite

Task 4. Importazione dei dati

Task 5. Queries

Query 1. Tutte le informazione dell'aeroporto di Pisa.

Query 2. Itinerari che arrivano a Pisa.

Query 3. Informazione spaziale (lat, lng) di partenza e di arrivo di tutti gli itinerari.

Query 4. Compagnie aeree che hanno almeno un itinerario con partenza da Pisa.

Query 5. Compagnie aeree senza nessuno itinerario.

Query 6. Aeroporti senza itinerario di arrivo.

Query 7. Nome e altitudine calcolata in metri degli aeroporti. (1 feet = 0.3048 meter)

Task 5. Queries

Query 8. Le città di partenza e di arrivo di ogni itinerario, e anche la differenza assoluta di altitudine (in metri) tra gli aeroporti ?

Query 9. L'Itinerario con maggiore differenza assoluta di altitude tra due aeroporti in metri.

Query 10. Nomi degli aeroporti e quantità di itinerari sia di partenza oppure di arrivo

Query 11. Nomi dei primi 10 aeroporti con più itinerari sia di partenza oppure di arrivo.

Query 12. I primi 10 itinerari più frequenti tra gli aeroporti.

Query 13. Compagnie aeree che offrono l'itinerario più frequente (Top 1).

Query 14. Calcolo di differenza tra l'altitudine di tutte le coppie di aeroporti.

Task 5. Queries

Query 1. Tutte le informazione dell'aeroporto di Pisa.

```
select *
from airport
where  city like ‘%Pisa%’
```

Query 2. Itinerari che arrivano a Pisa.

```
select r.*
from route r
inner join airport a on r.destinationAirportID == a.airportID
where a.city like '%Pisa%'
```

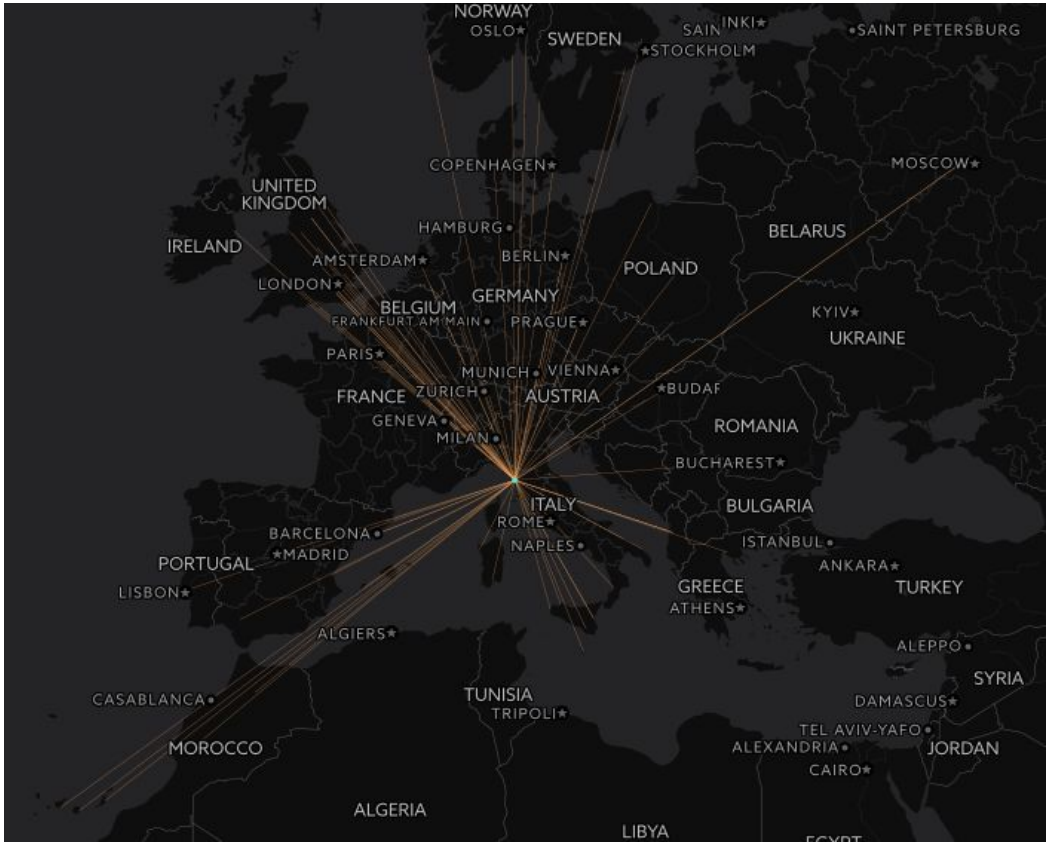
Task 5. Queries

Query 3. Informazione spaziale (lat, lng) degli aeroporti di partenza e di arrivo per tutti gli itinerari.

```
select
ao.name as airportSourceName
, ao.latitude as latitudeSource
, ao.longitude as longitudeSource
, ad.name as airportDestinationName
, ad.latitude as latitudeDestination
, ad.longitude as longitudeDestination
from route r
inner join airport ao on (r.sourceAirportID == ao.airportID)
inner join airport ad on (r.destinationAirportID == ad.airportID)
```

Task 5. Queries

Query 2. Itinerari che arrivano a Pisa.



Task 5. Queries

Query 4. Compagnie aeree che hanno almeno un itinerario con partenza da Pisa.

```
select distinct l.*
from route r
inner join airport a on r.sourceAirportID == a.airportID
inner join airline l on l.airlineID == r.airlineID
where a.city like '%Pisa%'
```

Task 5. Queries

Query 5. Compagnie aeree senza nessuno itinerario.

```
select l.*
from airline l
left join route r on (r.airlineID == l.airlineID)
where r.airlineID is null;

-- oppure

select *
from airline l
where not exists (
select r.airlineID
from route r where r.airlineID == l.airlineID
);
```

Task 5. Queries

Query 6. Aeroporti senza itinerario di arrivo.

```
select a.*
from airport a
left join route r on (r.destinationAirportID == a.airportID)
where r.destinationAirportID is null;
```

***Query 6.1. Nome degli aeroporti senza itinerario di arrivo e/o senza itinerario di partenza**

```
select sa.Name as SenzaArrivo,  sa.Name as SenzaPartenza
from  view_aeroporti_senza_itinerario_arrivo sa
full join view_aeroporti_senza_itinerario_partenza sp on sa.airportID == sp.airportID
```

Ricordando...

ESPRESSIONI

- Possono comparire nella SELECT
- Operandi
  - valori degli attributi
- Operatori (non standard)
  - operatori aritmetici +, -, *, % ed altri
  - funzioni matematiche log, exp, sin, …
  - funzioni su stringhe length, substring, …
  - funzioni su date e tempi

Task 5. Queries

Query 7. Nome e altitudine calcolata in metri degli aeroporti? (1 feet = 0.3048 meter).

```
select name,  altitude * 0.3048 as altitudeMeter from airport
```

Query 8. Le città di partenza e di arrivo di ogni itinerario, e anche la differenza assoluta di altitudine (in metri) tra gli aeroporti.

```
select ao.city as source, ad.city as destination, abs(ao.altitude - ad.altitude) * 0.3048
from route r
inner join airport ao on (r.sourceAirportID = ao.airportID)
inner join airport ad on (r.destinationAirportID = ad.airportID)
```

Query 9. L'Itinerario con maggiore differenza assoluta di altitude tra due aeroporti in metri.

```
(...)
order by abs(ao.altitude - ad.altitude) desc
limit 1
```

Task 5. Queries

Query 10. Nomi degli aeroporti e quantità di itinerari sia di partenza oppure di arrivo

```
select a.airportID, a.name, count(0)
from route r
inner join airport a on (r.destinationAirportID == a.airportID or r.sourceAirportID == a.airportID)
group by a.airportID, a.name
```

Query 11. Nomi dei primi 10 aeroporti con più itinerari sia di partenza oppure di arrivo.

```
(...)
order by count(0) desc
limit 10
```

Task 5. Queries

Query 12. I primi 10 itinerari più frequenti tra gli aeroporti.

```
select ao.airportID as airportIDSource, ao.name as parteza, ad.airportID as airportIDDestination,
ad.name as arrivo, count(0)
from route r
inner join airport ao on (r.sourceAirportID == ao.airportID)
inner join airport ad on (r.destinationAirportID == ad.airportID)
group by ao.airportID, ao.name, ad.airportID, ad.name
order by count(0) desc
limit 10
```

Task 5. Queries

Query 13. Compagnie aeree che offrono l'itinerario più frequente (Top 1).

```
select l.*
from (
    select ao.airportID as airportIDSource, ao.name as parteza, ad.airportID as airportIDDestination,
    ad.name as arrivo, count(0)
    from route r
    inner join airport ao on (r.sourceAirportID == ao.airportID)
    inner join airport ad on (r.destinationAirportID == ad.airportID)
    group by ao.airportID, ao.name, ad.airportID, ad.name
    order by count(0) desc
    limit 1) top1
inner join route r on (top1.airportIDDestination == r.destinationAirportID and top1.airportIDSource ==
r.sourceAirportID)
inner join airline l on l.airlineID = r.airlineID
```

Task 5. Queries

Query 13. Compagnie aeree che offrono l'itinerario più frequente (Top 1).

```
select l.*
from view_top1_itinerario_frequenti top1
inner join route r on (top1.airportIDDestination == r.destinationAirportID and top1.airportIDSource ==
r.sourceAirportID)
inner join airline l on l.airlineID = r.airlineID;

create view view_top1_itinerario_frequenti as
select ao.airportID as airportIDSource, ao.name as parteza, ad.airportID as airportIDDestination,
ad.name as arrivo, count(0)
from route r
inner join airport ao on (r.sourceAirportID == ao.airportID)
inner join airport ad on (r.destinationAirportID == ad.airportID)
group by ao.airportID, ao.name, ad.airportID, ad.name
order by count(0) desc limit 1;
```

Task 5. Queries

Query 14. Calcolo di differenza tra l'altitudine di tutte le coppie di aeroporti.

```
select a2.city, a1.city, a2.altitude - a1.altitude
from (select * from airport limit 5) a1
cross join (select * from airport limit 5) a2
order by a2.altitude - a1.altitude
```