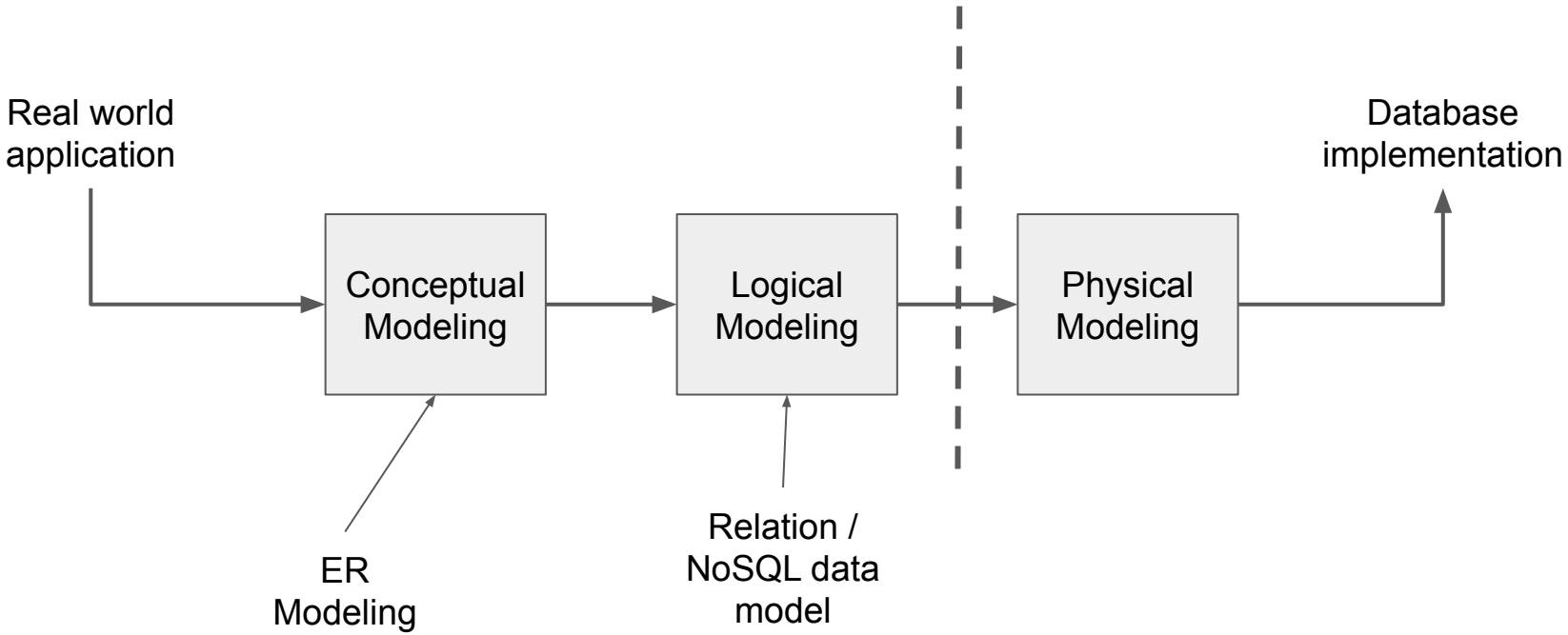
Database and Big Data Systems

Lab 3

Today

- Recap
- SQL

Recap


Real world application
Conceptual Modeling
Logical Modeling
Physical Modeling
Database implementation
ER Modeling
Relation / NoSQL data model

SQL

- Declarative language
- Data definition
  - Create database, tables
- Data manipulation
  - Insert, update, delete
- Query
  - Select … From … Where

SQL - Data Definition

- Create database and tables

Payroll (UserID, Name, Job, Salary)

```
create table Payroll (
    UserID integer,
    Name varchar(100),
    Job varchar(100),
    Salary integer
);
```

```
create database if not exists university;
```

Case insentive (except for Table name).

But *please* don't capitalize everything.

SQL - Data Manipulation

- Add new data into table

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000
001	Anh	Prof	10000
002	Cyrille	TA	10000

```
insert into Payroll values (001, "Anh", “Prof”, 10000);
insert into Payroll values (002, "Cyrille", “TA”, 10000);
```

=

```
insert into Payroll values (001, "Anh", “Prof”, 10000),
                           (002, "Cyrille", “TA”, 10000);
```

SQL - Query

1. Take product of input relation R1, R2,..
2. Apply selection condition
3. Take specific column Col1, Col2, ...

Today

- Recap
- SQL

SQL

- MySQL doesn’t support set subtraction
- Write a query that returns R(**a**) - S(**a**)
  - Table R has one attribute a (primary key)
  - Table S has one attribute a (primary key)

SQL

- MySQL doesn’t support set subtraction
- Write a query that returns R(**a**) - S(**a**)
  - Table R has one attribute a (primary key)
  - Table S has one attribute a (primary key)

```
Select a from R
Where a not in (select a from S);

Select a from R
Where not exist (select a from S where R.a = S.a);


select R.a from R left join S on R.a = S.a where S.a is null;
```

SQL

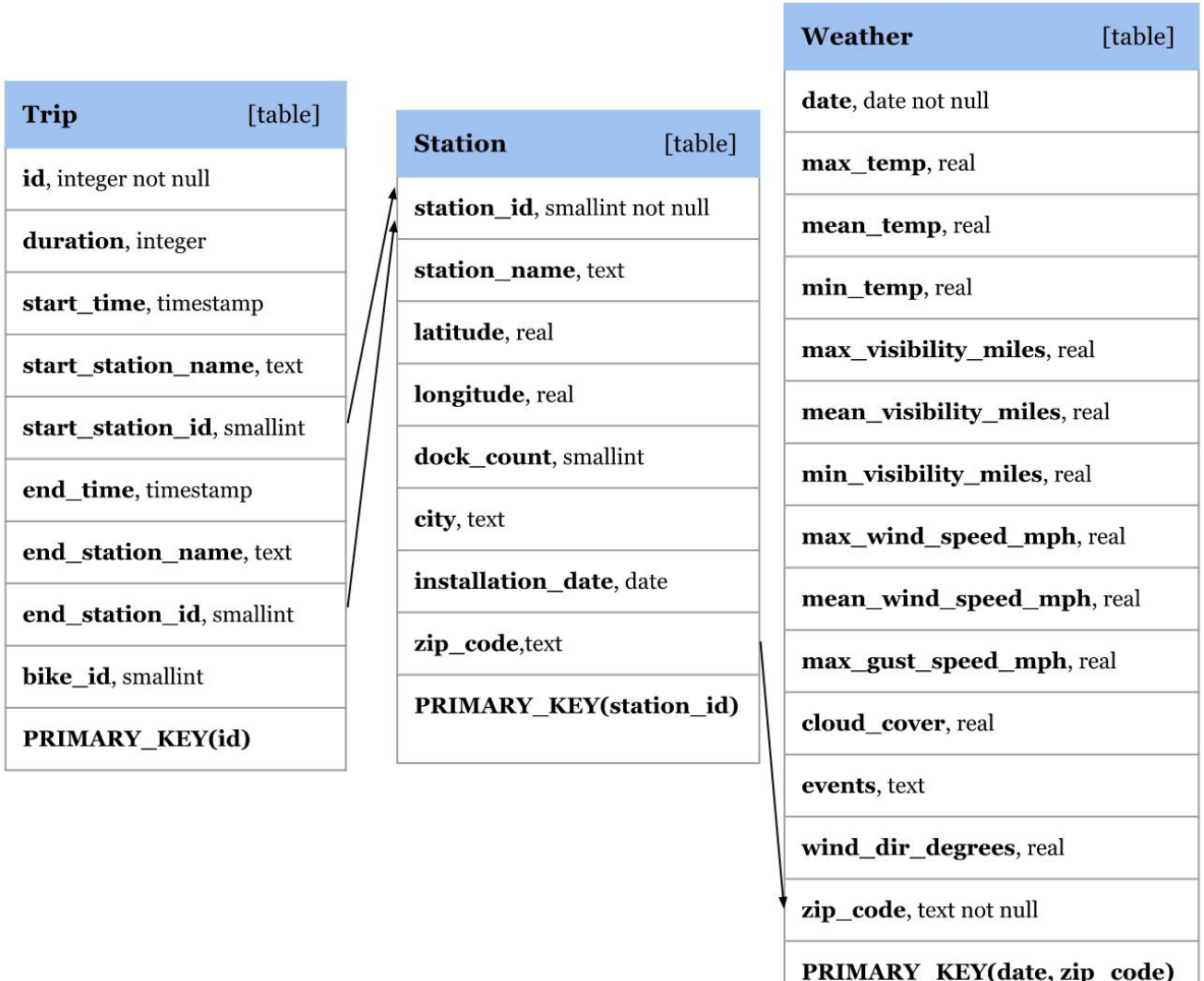
- Launch an instance from the following image (US-East1)

  ami-09f7eb2d6268ba83a

- If you are not using AWS, a local Linux/MacOS machine with MySQL installed works too.

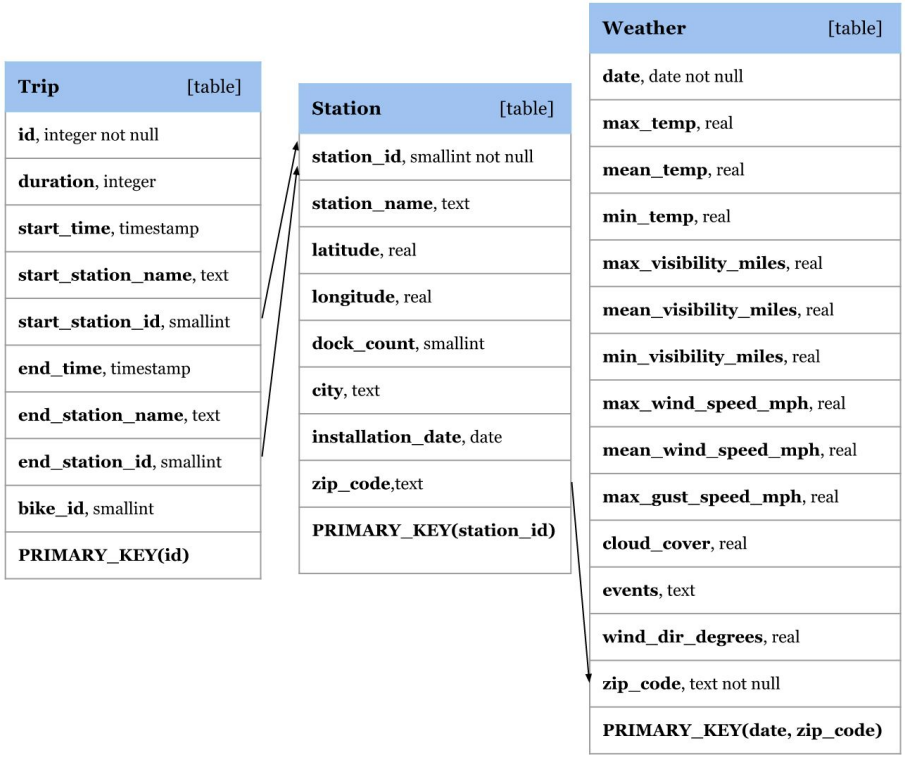
SQL

- Data schema
- Exercises: create the tables



SQL

- Insert new data in
- Check for constraints
  - Primary key constraint
  - Foreign key constraint
- Delete tables and database



SQL

- Load data in with /home/ubuntu/setup.sql
- If you are using a local machine, you may download the setup.sql files (v5.7 and v8) here.

SQL

- [Q1] Count the number of cities (no duplicates)
- [Q2] Count the number of stations in each city. Output the city name, and station count.
- [Q3] Count the number of self-loop trips. A self-loop trip is one that starts and ends at the same station.

SQL

- How to create a temporary table from some SQL query
  - `create temporary table <name> <SQL query>`
  - Useful for replacing nested queries
- [Q4] Print the ratio of self-loop trips over all trips.
- [Q5] Find the most popular city, in terms of percentage of trips belonging to the city.

SQL

- [Q6] Find all the bikes (their bike_id) that have been to more than 1 city. A bike has been to a city if its start or end station of one of its trips is in this city.
- [Q7] List the bikes (their bike_id) that have never been to Japantown station.