Tutorial: Database Design

> Designed by ZHU Yueming. A small part of descriptions of basic concepts in this tutorial are borrowed from the Stephane Faroult's Slide and Wikipedia.

Experimental Objective

- Understand basic constraints in database design
- Understand 3 normal form of database design
- Learn how to design a database according to the requirement document
- Understand basic DDL (Data Definition Language) language

Part 1. Constraints

Constraints are declarative rules that the DBMS apply to ensure the integrity of data. DBMS will check constraints every time new data is added, changed or deleted, to prevent any inconsistency. Any operation that violates a constraint fails and returns an error.

1. NOT NULL

If you want one column cannot be null, you can indicate the column by `not null`.

Sample DDL language.

```
create table if not exists stations
(
    station_id integer not null
        constraint stations_pkey
            primary key,
    english_name varchar(80) not null
        constraint stations_uq_1
            unique,
    chinese_name varchar(10) not null
        constraint stations_uq_2
            unique,
    district varchar(20),
    latitude double precision,
    longitude double precision
);
```

Sample graph:

station_id	english_name	chinese_name	district	latitude	longitude
156	Huangmugang	黄木岗	Futian	22.56111	114.09306
157	Bagualing	八卦岭	Futian	22.56806	114.10111
158	Hongling North	红岭北	Luohu	22.56833	114.11083
159	Sungang	笋岗	Luohu	22.56972	114.11778
160	Honghu	洪湖	Luohu	22.57222	114.12944
161	Wenti Park	文体公园	Nanshan	<null>	<null>
162	Hanghai Road	航海路	Nanshan	<null>	<null>
163	Zhenhai Road	振海路	Nanshan	<null>	<null>
164	Linhai Road	临海路	Nanshan	<null>	<null>
165	Qianhai Road	前海路	Nanshan	<null>	<null>
166	Lixiang	荔香	Nanshan	<null>	<null>
167	Nanyou	南油	Nanshan	<null>	<null>

can be null

2. UNIQUE

Each value in the specified column(s) is unique.

- Unique for one column

  Sample DDL language.

```
create table if not exists stations
(
    station_id integer not null
        constraint stations_pkey
            primary key,
    english_name varchar(80) not null
        constraint stations_uq_1
            unique,
    chinese_name varchar(10) not null
        constraint stations_uq_2
            unique,
    district varchar(20),
    latitude double precision,
    longitude double precision
);
```

  Sample graph:

english_name	chinese_name	district
Luohu	罗湖	Luohu
Guomao	国贸	Luohu
Laojie	老街	Luohu
Grand Theater	大剧院	Luohu
Science Museum	科学馆	Futian
Huaqiang Rd	华强路	Futian
Gangxia	岗厦	Futian
Convention and Exhibition…	会展中心	Futian
Shopping Park	购物公园	Futian
Xiangmihu	香蜜湖	Futian
Chegongmiao	车公庙	Futian
Zhuzilin	竹子林	Futian
Qiaocheng East	侨城东	Futian
OCT	华侨城	Nanshan
Window of the World	世界之窗	Nanshan

unique

not unique

- Unique for multiple columns

  Sample DDL Language

```
create table test_unique (
    id serial primary key,
    english_name varchar(80) not null,
    line_id integer not null,
    line_color varchar(20) not null,
    district varchar(20),
    constraint uq unique (english_name, line_id)
);
```

  Sample graph:

english_name	line_id	line_color	district
Bao'an Center	5	DarkOrchid	Bao'an
Buji	5	DarkOrchid	Longgang
Buji	3	DeepSkyBlue	Longgang
Chegongmiao	11	Purple	Futian
Chegongmiao	1	Green	Futian
Chegongmiao	9	DimGray	Futian
Chegongmiao	7	MediumBlue	Futian
Children's Palace	3	DeepSkyBlue	Futian
Children's Palace	4	Red	Futian
Civic Center	4	Red	Futian
Civic Center	2	Orange	Futian
Convention and Exhi…	4	Red	Futian
Convention and Exhi…	1	Green	Futian

3. PRIMARY KEY

Primary key specifies the main key for the table, which is:

- Mandatory (the additional NOT NULL doesn't hurt but is redundant)
- Unique (no duplicates allowed in the column)

Sample DDL Language

```
create table if not exists lines
(
    line_id integer not null
            primary key,
    line_color varchar(20) not null
            unique,
    opening integer,
    latest_extension integer,
    operator varchar(30)
);
```

Sample graph:

line_id	line_color	opening	latest_extension	operator
1	Green	2004	2011	Shenzhen Metro Corporation
2	Orange	2010	2011	Shenzhen Metro Corporation
3	DeepSkyBlue	2010	2011	Shenzhen Metro No.3 Line
4	Red	2004	2011	MTR Corporation
5	DarkOrchid	2011	<null>	Shenzhen Metro Corporation
6	LightSeaGreen	2019	<null>	<null>
7	MediumBlue	2016	<null>	Shenzhen Metro Corporation
8	Violet	2020	<null>	<null>
9	DimGray	2016	<null>	Shenzhen Metro Corporation
10	HotPink	2019	<null>	<null>
11	Purple	2016	<null>	Shenzhen Metro Corporation
12	Amethyst	<null>	<null>	<null>
13	Coral	<null>	<null>	<null>
14	DarkGray	<null>	<null>	<null>
15	LimeGreen	<null>	<null>	<null>
16	CornFlowerBlue	<null>	<null>	<null>

primary key

4. FOREIGN KEY

Foreign key indicates that the column must reference a key (Only primary keys and columns declared as UNIQUE) of another table.

- Constraints are used to prevent actions that break the connection between tables
- Constraints also prevent illegal data from being inserted into the column.

Sample DDL Language: we add a foreign key constraint `station_id` in table `connections`, which references the primary key `station_id` in table `stations`

```
create table connections
(
    station_id integer      not null
        constraint connections_fk
            references stations(station_id),
    connection varchar(100) not null,
    constraint connections_pk
        primary key (station_id, connection)
);
```

其他表

Sample graph:

stations

station_id	english_name	chinese_name	district	latitude	longitude
1	Luohu	罗湖	Luohu	22.53111	114.11833
2	Guomao	国贸	Luohu	22.54	114.11889
3	Laojie	老街	Luohu	22.54444	114.11639
4	Grand Theater	大剧院	Luohu	22.54472	114.10333

Foreign key

connections

station_id	connection
1	MTR via Lo Wu [thru border check]
1	(9 via Renmin South)
1	China Railway
1	China Railway High-speed Shenzhen
3	3
4	2
4	(9 via Hongling South)

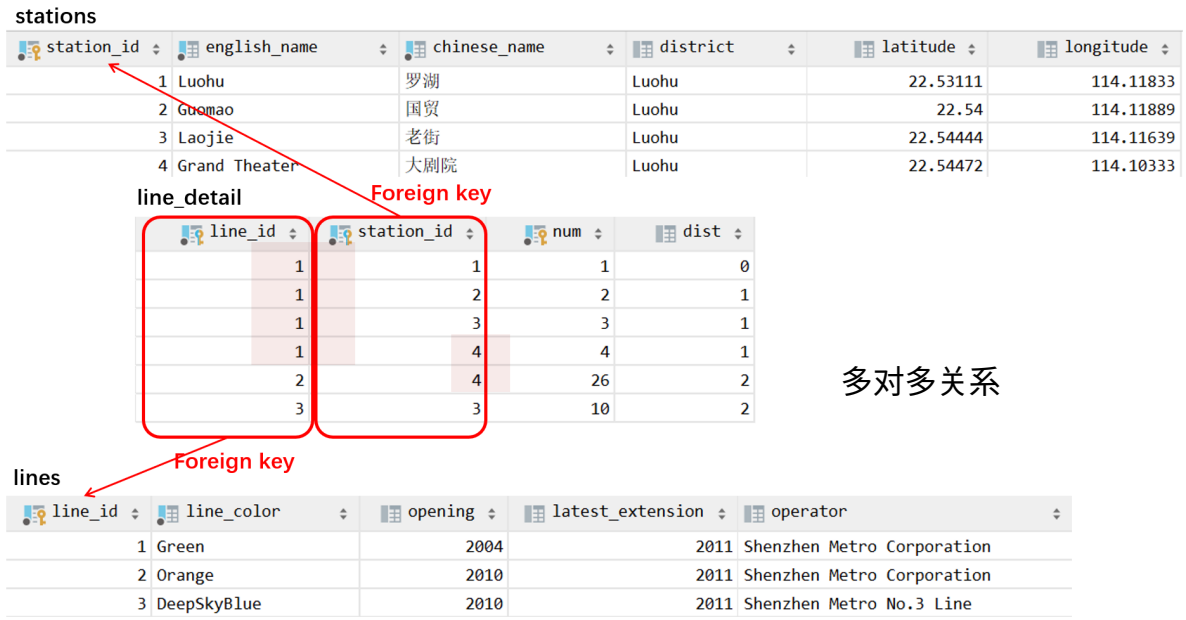
In this case, we cannot insert any data in connections if the inserted station_id is not appeared in the column station_id in stations table because of the foreign key. For example

```
insert into connections (station_id, connection) values (10086,'CSE');
```

The result would be:

```
[23503] ERROR: insert or update on table "connections" violates foreign key
constraint "connections_fk" 详细：Key (station_id)=(10086) is not present in table
"stations".
```

Another sample graph below represents one table can have multiple foreign keys.



Part 2. Normal Form

1 NF

A relation is in first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain.

Comparison 1:

The first one doesn't satisfy 1NF, because the attribute is not atomic. More specifically, the name can be split to English name and Chinese name, location can be split to longitude and latitude.

First:

station_id	name	location
1	Luohu(罗湖)	114.11833 , 22.53111
2	Guomao(国贸)	114.11889 , 22.54
3	Laojie(老街)	114.11639 , 22.54444
4	Grand Theater(大剧院)	114.10333 , 22.54472
5	Science Museum(科学馆)	114.08972 , 22.54333
6	Huaqiang Rd(华强路)	114.07889 , 22.54306
7	Gangxia(岗厦)	114.06306 , 22.53778
8	Convention and Exhibition Center Station(会展中心)	114.05472 , 22.5375
9	Shopping Park(购物公园)	114.05472 , 22.53444
10	Xiangmihu(香蜜湖)	114.034 , 22.5417

Improved: 分开

station_id	english_name	chinese_name	longitude	latitude
1	Luohu	罗湖	114.11833	22.53111
2	Guomao	国贸	114.11889	22.54
3	Laojie	老街	114.11639	22.54444
4	Grand Theater	大剧院	114.10333	22.54472
5	Science Museum	科学馆	114.08972	22.54333
6	Huaqiang Rd	华强路	114.07889	22.54306
7	Gangxia	岗厦	114.06306	22.53778
8	Convention and Exhibition Cent…	会展中心	114.05472	22.5375
9	Shopping Park	购物公园	114.05472	22.53444
10	Xiangmihu	香蜜湖	114.034	22.5417

Comparison 2:

The first one doesn't satisfy 1NF, because the stations has more value of the domain, more specifically,

column station1 and station2 are similar, so that we should merge them into one column.

First:

line_id	line_color	station1	station2
1	Green	Bao'an Stadium	Pingzhou
2	Orange	Chiwan	Shekou Port
3	DeepSkyBlue	Danzhutou	<null>

Improved:

line_id	line_color	stations
1	Green	Bao'an Stadium
1	Green	Pingzhou
2	Orange	Chiwan
2	Orange	Shekou Port
3	DeepSkyBlue	Danzhutou

2 NF

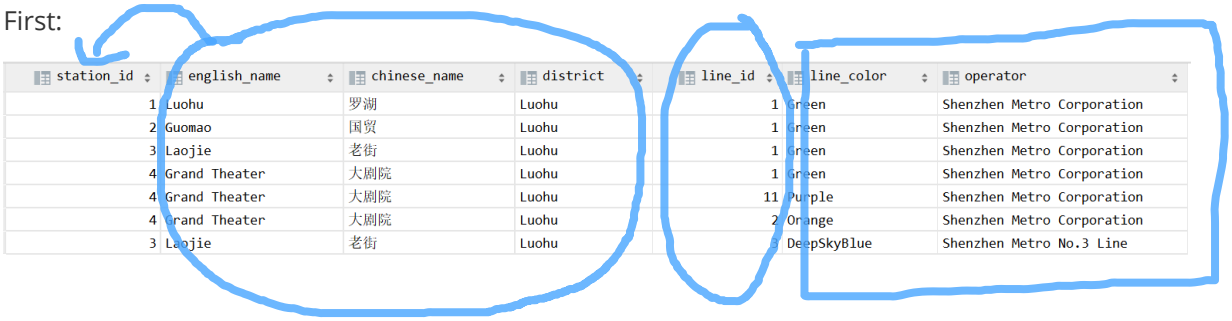
A relation satisfying 2NF must:

- be in 1NF
- not have any non-prime attribute that is dependent on any proper subset of any candidate key of the relation. A non-prime attribute of a relation is an attribute that is not a part of any candidate key of the relation.

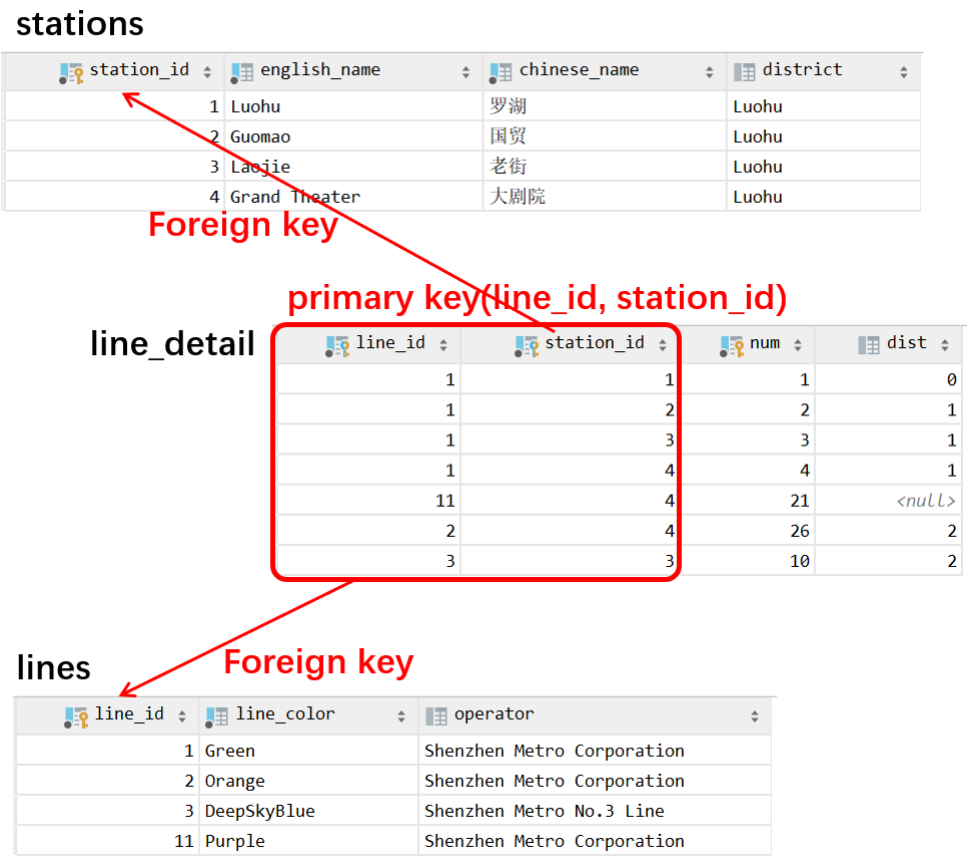
The primary keys in the first table are station_id and line_id, but the first one is a awful design because the columns line_color and operator are not related to station_id(primary key) while english_name, chinese_name and district are not related to line_id

If we want to describe **many-to-many relationship** between two entities, we need to separate those two entities into two tables and then add a relation table to describe the relationship itself (as in the second case).

First:



Improved:



stations

station_id	english_name	chinese_name	district
1	Luohu	罗湖	Luohu
2	Guomao	国贸	Luohu
3	Laojie	老街	Luohu
4	Grand Theater	大剧院	Luohu

line_detail

line_id	station_id	num	dist
1	1	1	0
1	2	2	1
1	3	3	1
1	4	4	1
11	4	21	<null>
2	4	26	2
3	3	10	2

lines

line_id	line_color	operator
1	Green	Shenzhen Metro Corporation
2	Orange	Shenzhen Metro Corporation
3	DeepSkyBlue	Shenzhen Metro No.3 Line
11	Purple	Shenzhen Metro Corporation

3 NF

A relation satisfying 2NF must:

- be in 2NF
- all the attributes in a table are determined only by the candidate keys of that relation and not by any non-prime attributes.

We can see from the figure that the column bus_line is not related to the primary key (station_id). The column is more suitable for describing bus_line. In this case, we only use the id of station_id, which serves as a foreign key referencing the station_id table, in the bus_lines table.

If we want to describe **one-to-many relationship**, e.g. A only has one B, but B could have more than one A, we can design a foreign key in A table referencing the primary key in B table.

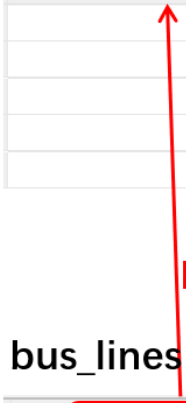
First:

station_id	chinese_name	english_name	district	bus_line
21	鲤鱼门	Liyumen	Nanshan	B796
21	鲤鱼门	Liyumen	Nanshan	M343
21	鲤鱼门	Liyumen	Nanshan	M349
26	坪洲	Pingzhou	Bao'an	M250
61	安托山	Antuo Hill	Futian	374
61	安托山	Antuo Hill	Futian	B733
120	临海	Linhai	Nanshan	B828
121	宝华	Baohua	Bao'an	B828
121	宝华	Baohua	Bao'an	M235

Improved:

stations

station_id	chinese_name	english_name	district
21	鲤鱼门	Liyumen	Nanshan
26	坪洲	Pingzhou	Bao'an
61	安托山	Antuo Hill	Futian
120	临海	Linhai	Nanshan
121	宝华	Baohua	Bao'an



bus_lines

station_id	bus_line
21	B796
21	M343
21	M349
26	M250
61	374
61	B733
120	B828
121	B828
121	M235

Part 3. Sample data definition language （DDL)

1. Create Table Example

- First table represents ticket.

```
create table if not exists ticket
(
    id           serial primary key,
    train_number varchar(10) not null unique,
    depart_city  varchar     not null,
    arrival_city varchar     not null,
    depart_time  time,
    price        numeric check (price > 0)
);
```

- Second table represents the customer.

```
create table customer
(
    id           serial,
    username     varchar(10) not null unique,
    password     varchar,
    phone_number varchar(11)
);
```

- Final table represents the records of booking ticket for customer, which can serve as a relation table of ticket and customer.

```
create table booking_record
(
    id          serial primary key,
    ticket_id   integer,
    customer_id     integer,
    date        date not null,
    depart_city varchar,
    arrive_city varchar,
    constraint fk_ticket foreign key (ticket_id) references ticket(id)
);
```

2. Alter Table Example

1) add not null constraint

General Syntax:

```
ALTER TABLE [ IF EXISTS ] [ ONLY ] table_name
    ALTER [ COLUMN ] column_name { SET | DROP } NOT NULL
```

Example:

```
alter table customer alter password set not null;
```

2) add primary key

General Syntax:

```
ALTER TABLE table_name
ADD CONSTRAINT MyPrimaryKey PRIMARY KEY (column1, column2...);
```

Example:

```
alter table customer add constraint pk_customer_id primary key (id);
```

3) add foreign key

General Syntax:

```
ALTER TABLE table_name
ADD CONSTRAINT ForeignKey FOREIGN KEY (column) REFERENCES table2 (column);
```

Example:

```
alter table booking_record add constraint fk_customer foreign key (customer_id)
references customer(id);
```

4) change data type of column

General Syntax:

```
ALTER TABLE table_name ALTER COLUMN column_name TYPE datatype;
```

Example:

```
alter table customer alter column phone_number type varchar;
```

5) add one column

General Syntax:

```
ALTER TABLE table_name ADD COLUMN column_name TYPE datatype;
```

Example:

```
alter table ticket add column seat_type varchar(2);
```

6) drop one column

General Syntax:

```
ALTER TABLE table_name DROP COLUMN column_name;
```

Example:

```
alter table booking_record drop column depart_city;
alter table booking_record drop column arrive_city;
```

7) drop constraints

General Syntax:

```
ALTER TABLE table_name DROP CONSTRAINT constraint_name;
```

Example:

```
alter table booking_record add constraint unique_columns unique
(customer_id,ticket_id,date);
alter table booking_record drop constraint unique_columns;
```

8) rename

General Syntax:

```
ALTER TABLE table_name RENAME TO new_name;
ALTER TABLE table_name RENAME COLUMN column_name TO new_column
```

Example:

```
alter table booking_record rename column date to booking_date;
```

9). add check constraints

General Syntax:

```
ALTER TABLE table_name ADD CONSTRAINT constraint_name CHECK (CONDITION);
```

Example:

```
alter table customer add constraint check_phone check ( length(phone_number)>=11
);
```

3. Drop Table Example

- Firstly, you drop the table, which has been referenced by an foreign key of other table. Try the query below

  ```
  drop table customer;
  ```

  The result would be:

  ```
  [2BP01] 错误: 无法删除 表 customer 因为有其它对象倚赖它 详细：在表 booking_record上
  的约束fk_customer 倚赖于 表 customer 建议：使用 DROP .. CASCADE 把倚赖对象一并删除.
  ```

- Solution 1. Drop table customer and all its foreign key constraints in other tables.

  ```
  drop table customer cascade;
  ```

- Solution 2. Drop the all other tables which have foreign key constraint that related to the current table first, and then drop current table.

  ```
  drop table booking_record;
  drop table customer;
  ```

Part 4. Exercise

Design a simple database which contains three tables as follows:

- student (name, student_id, department, gender)
- department (name, location, website)
- course (name, course_number, department, credit)

Other requirements describe as follows: The relationship between Student and Course is many-to-many. The relationship between Course and Department is many-to-one. The relationship between Student and Department is many-to-one.

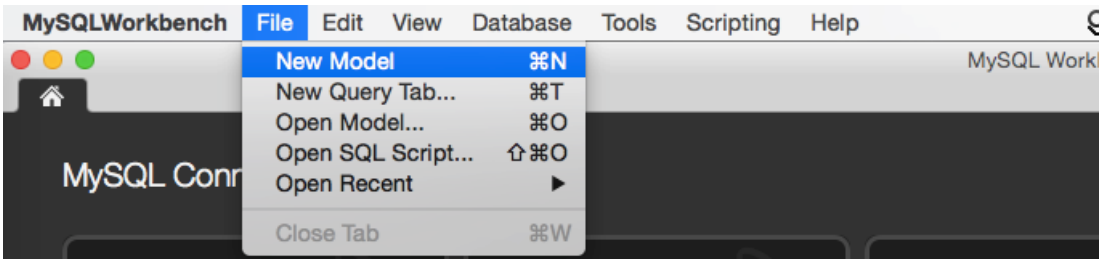
Please design a simple database that can match all requirements above by DDL language in postgres.

Part 5. Design ER diagram by Mysql workbench (Additional)

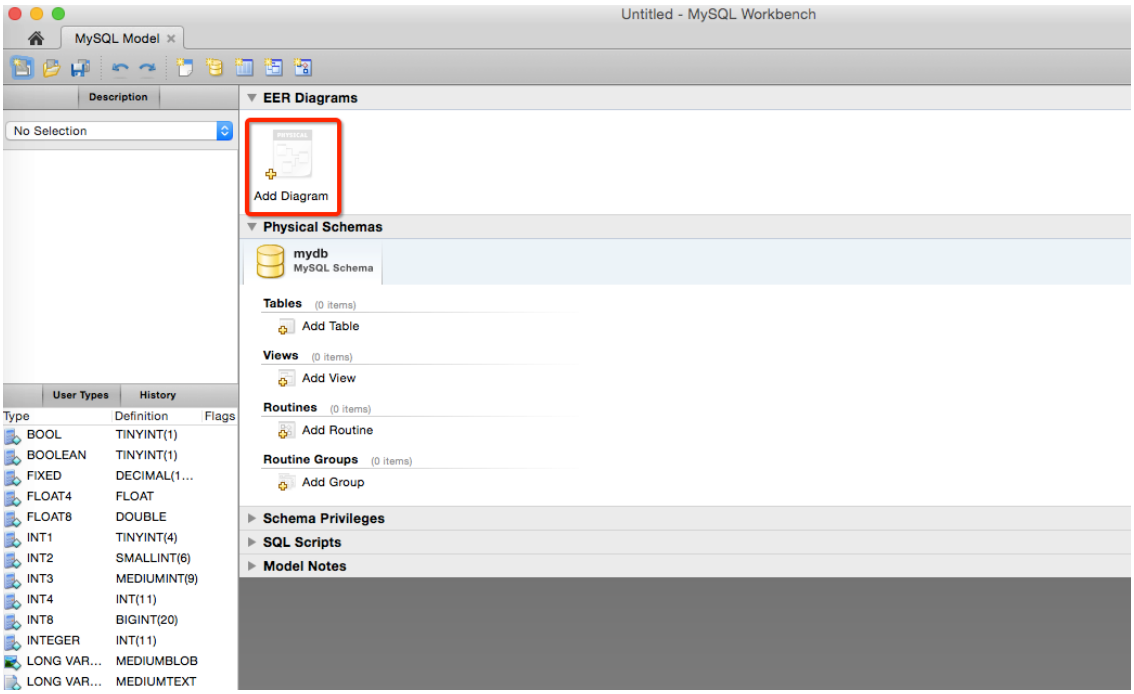
Here to download.

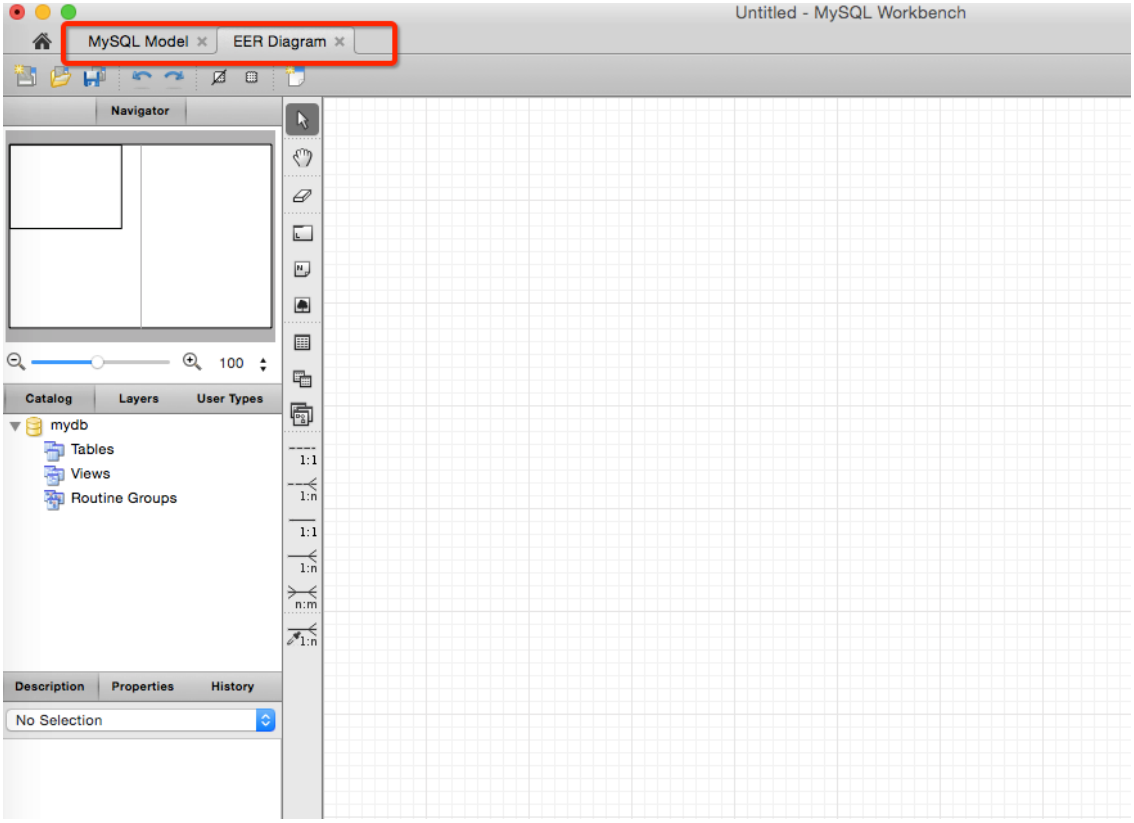
How to use it?

- Step 1 File—New Model

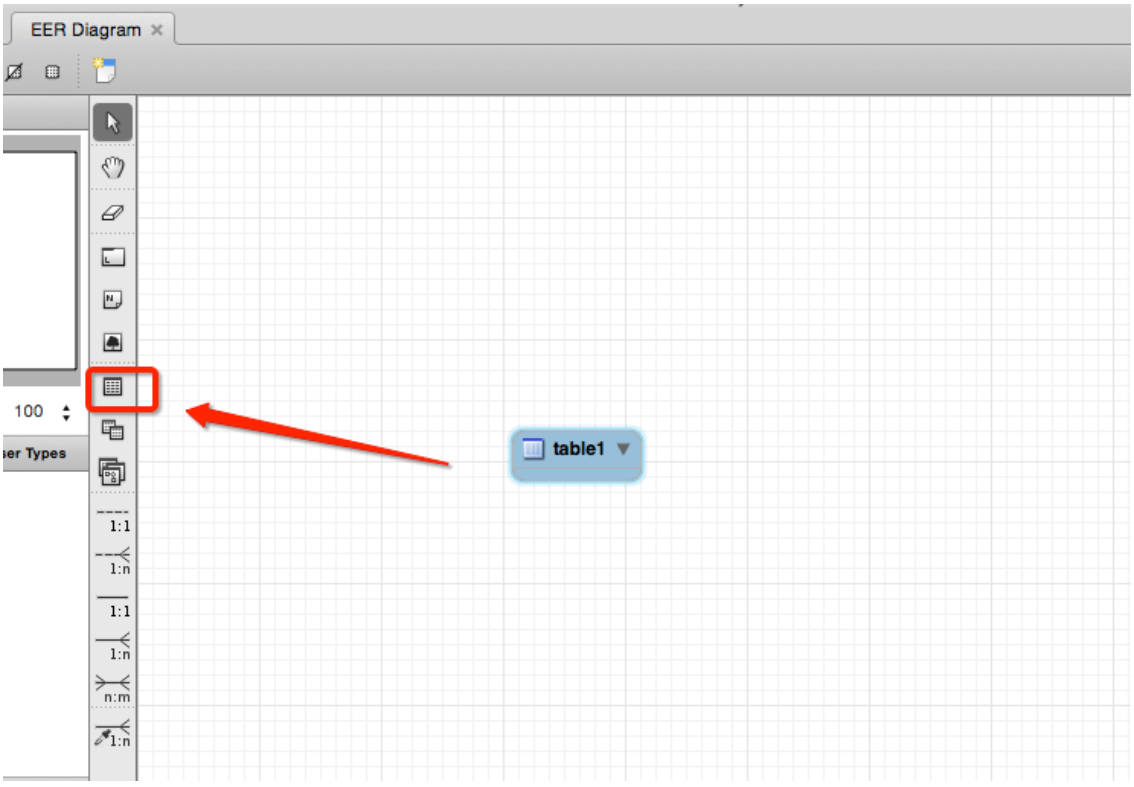


- Step 2 Double click Add diagram

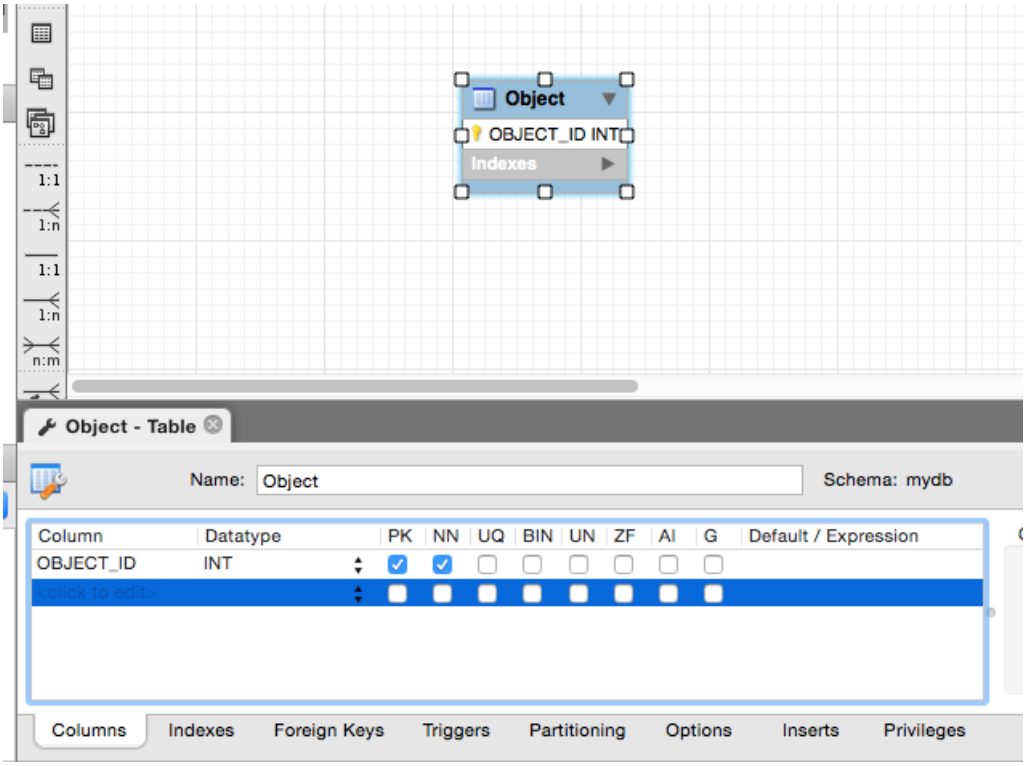




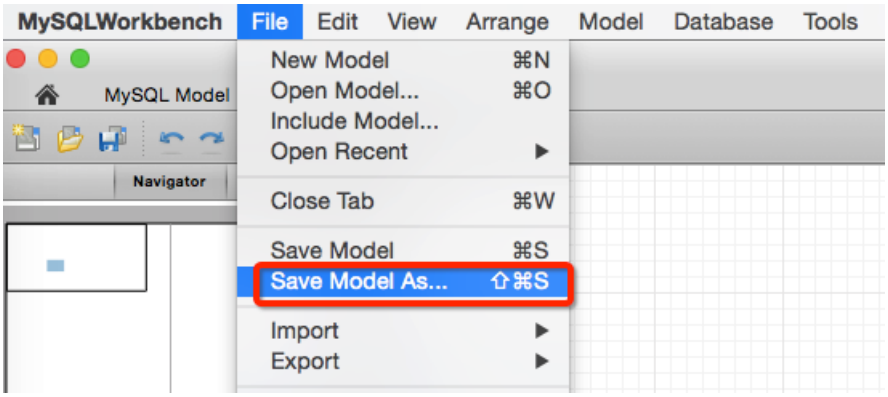
- Step 3 Add table to ER diagram by clicking this button



- Step 4 Double click table in ER diagram and edit it. Give it a name and add other columns.



- Step 5 When you finish editing your ER diagram, do not forget to save it



Tips: I suggest you edit relationship between tables by setting foreign key from one table to another. Don’t create relationships by adding lines directly in ER diagram.