

LAPORAN TUGAS 1

DATA WAREHOUSE DAN BUSINESS INTELLIGENCE



Oleh :

Jessica Rameylin Gultom

181402114

TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN 2021

Module 3 Problems

The problems use the *Customer*, *Facility*, and *Location* tables of the intercollegiate athletic database. The *Customer* table contains clients who initiate event requests. The *Facility* table contains available facilities. The *Location* table contains several locations inside facilities. The primary keys of the tables are *CustNo* for *Customer*, *FacNo* for *Facility*, and *LocNo* for *Location*.

Customer

custno	custname	address	Internal	contact	phone	city	state	zip
C100	Football	Box 352200	Y	Mary Manager	6857100	Boulder	CO	80309
C101	Men's Basketball	Box 352400	Y	Sally Supervisor	5431700	Boulder	CO	80309
C103	Baseball	Box 352020	Y	Bill Baseball	5431234	Boulder	CO	80309
C104	Women's Softball	Box 351200	Y	Sue Softball	5434321	Boulder	CO	80309
C105	High School Football	123 AnyStreet	N	Coach Bob	4441234	Louisville	CO	80027

Facility

facno	facname
F100	Football stadium
F101	Basketball arena
F102	Baseball field
F103	Recreation room

Location

locno	facno	locname
L100	F100	Locker room
L101	F100	Plaza
L102	F100	Vehicle gate
L103	F101	Locker room
L104	F100	Ticket Booth
L105	F101	Gate
L106	F100	Pedestrian

TUGAS 1

1. Write a CREATE TABLE statement for the *Customer* table. Choose data types appropriate for the DBMS used in your course. All columns are required (not null).

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (2.11 sec)
```

```
mysql> create database datawarehouse;
Query OK, 1 row affected (2.04 sec)
```

```
mysql> use datawarehouse;
Database changed
mysql> create table Customer (
  -> custno CHAR (11) primary key not null,
  -> custname VARCHAR(50) not null,
  -> address VARCHAR(50) not null,
  -> internal BOOLEAN not null,
  -> contact VARCHAR(50) not null,
  -> phone VARCHAR(50) not null,
  -> city VARCHAR(50) not null,
  -> state CHAR(10) not null,
  -> zip CHAR(10) not null
  -> );
Query OK, 0 rows affected (0.94 sec)
```

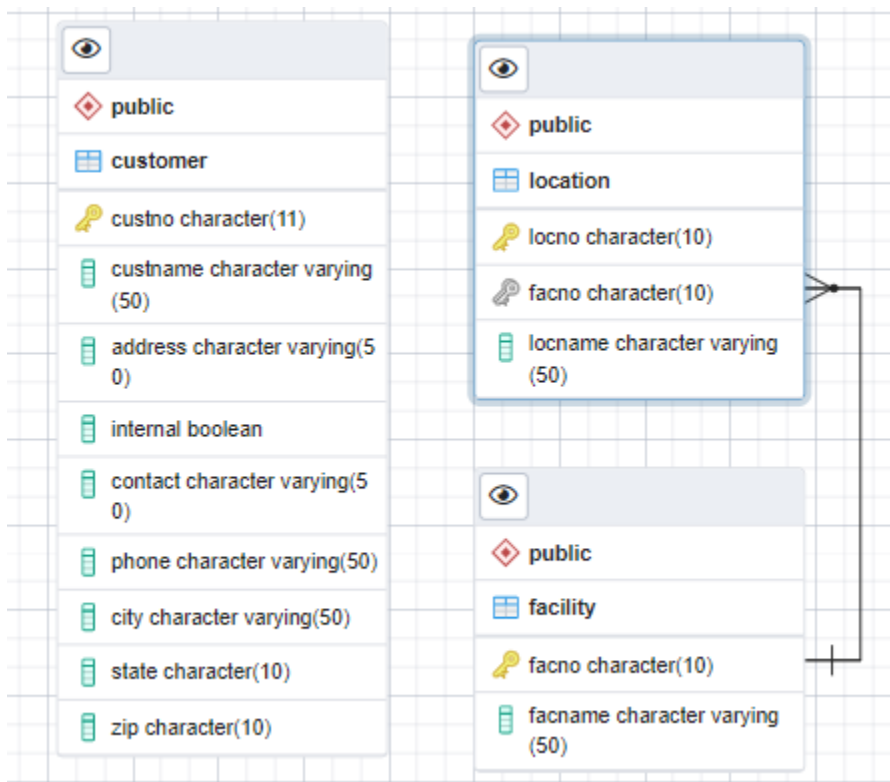
2. Write a CREATE TABLE statement for the *Facility* table. Choose data types appropriate for the DBMS used in your course. All columns are required (not null).

```
mysql> create table Facility (
  -> facno CHAR(10) primary key not null,
  -> facname VARCHAR(50) not null
  -> );
Query OK, 0 rows affected (0.51 sec)
```

3. Write a CREATE TABLE statement for the *Location* table. Choose data types appropriate for the DBMS used in your course. *LocName* column is required (not null).

```
mysql> create table Location (
-> locno CHAR(10) primary key,
-> facno CHAR(10),
-> locname VARCHAR(50) not null
-> );
Query OK, 0 rows affected (0.94 sec)
```

4. Identify the foreign key(s) and 1-M relationship(s) among the *Customer*, *Facility*, and *Location* tables. For each relationship, identify the parent table and the child table.



- Hubungan antara Facility dan Location adalah 1-1 relationship. Dimana tabel facility sebagai parent tabel dan tabel Location sebagai child tabel. Ini dibuktikan dengan adanya attribut 'facno' yang menjadi primary key pada tabel Facility dan 'facno' yang menjadi foreign key pada tabel Location.
5. Extend your CREATE TABLE statement from problem (3) with referential integrity constraints.

```
mysql> create table Location (
  -> locno CHAR(10) primary key not null,
  -> facno CHAR(10) not null,
  -> locname VARCHAR(50) not null,
  -> CONSTRAINT facno FOREIGN KEY(facno) REFERENCES Facility (facno)
  -> );
Query OK, 0 rows affected (5.36 sec)
```

6. From examination of the sample data and your common understanding of scheduling and operation of events, are null values allowed for the foreign key in the *Location* table? Why or why not? Extend the CREATE TABLE statement in problem (5) to enforce the null value restrictions if any.

```
mysql> create table Location (
  -> locno CHAR(10) not null,
  -> facno CHAR(10) not null,
  -> locname VARCHAR(50) not null,
  -> Constraint PK_Location PRIMARY KEY(locno),
  -> CONSTRAINT FK_Facno FOREIGN KEY(facno) REFERENCES Facility(facno)
  -> );
Query OK, 0 rows affected (3.48 sec)
```

7. Extend your CREATE TABLE statement for the *Facility* table (problem 2) with a unique constraint for *FacName*. Use an external named constraint clause for the unique constraint.

```
mysql> create table Facility (
  -> facno char(10) not null,
  -> facname Varchar(30) not null,
  -> CONSTRAINT PK_Facility PRIMARY KEY (facno),
  -> CONSTRAINT Unique_facname UNIQUE(facname)
  -> );
Query OK, 0 rows affected (2.05 sec)
```