Nama : Alvin Daeli

NIM : 181402054

Mata Kuliah : Data Warehouse dan Bisnis Intelligence

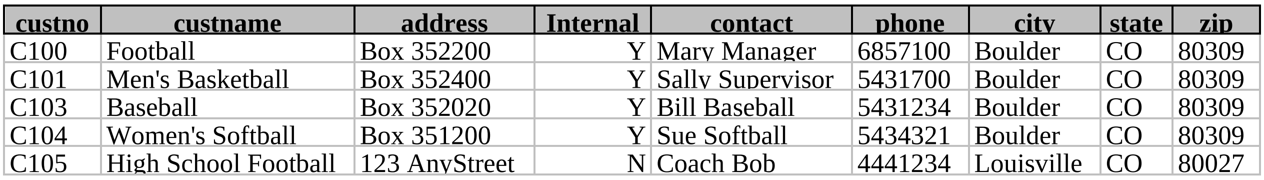
Kom : C

Tugas 1

Module 3 Problems

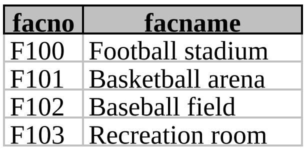
The problems use the *Customer*, *Facility*, and *Location* tables of the intercollegiate athletic database. The *Customer* table contains clients who initiate event requests. The *Facility* table contains available facilities. The *Location* table contains several locations inside facilities. The primary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

Customer



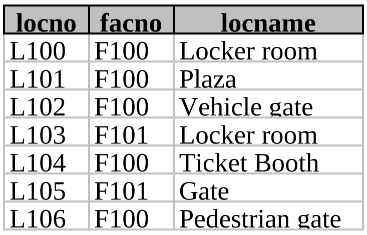
custno	custname	address	Internal	contact	phone	city	state	zip
C100	Football	Box 352200	Y	Mary Manager	6857100	Boulder	CO	80309
C101	Men's Basketball	Box 352400	Y	Sally Supervisor	5431700	Boulder	CO	80309
C103	Baseball	Box 352020	Y	Bill Baseball	5431234	Boulder	CO	80309
C104	Women's Softball	Box 351200	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 gate

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).

   Answer :

   ```
   CREATE TABLE Customer (CustNo VARCHAR(4) NOT NULL, CustName VARCHAR(32)
   NOT NULL, Address VARCHAR(50) NOT NULL, Internal CHAR(1) NOT NULL,
   Contact VARCHAR(64) NOT NULL, Phone VARCHAR(7) NOT NULL, City
   VARCHAR(32) NOT NULL, State VARCHAR(2) NOT NULL, Zip VARCHAR(5) NOT
   NULL, CONSTRAINT PK_CUSTOMER PRIMARY KEY (CustNo) );
   ```

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).

   Answer :

   ```
   CREATE TABLE Facility (FacNo VARCHAR(4) NOT NULL, FacName VARCHAR(32)
   NOT NULL, CONSTRAINT PK_FACILITY PRIMARY KEY (FacNo) );
   ```

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).

   Answer :

   ```
   CREATE TABLE Location (LocNo VARCHAR(4) NOT NULL, FacNo VARCHAR(4),
   LocName VARCHAR(32) NOT NULL, CONSTRAINT PK_LOCATION PRIMARY KEY
   (LocNo));
   ```

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.

   Answer :

   Facility table has a 1-M relationship to ***Location*** table where ***FacNo*** in ***Facility*** are **primary key** while ***FacNo*** in ***Location*** table are **foreign key**. For this relationship, ***Facility*** is the parent table while ***Location*** is the child table.

5. Extend your CREATE TABLE statement from problem (3) with referential integrity constraints.

   Answer :

   ```
   CREATE TABLE Location (LocNo VARCHAR(4) NOT NULL, FacNo VARCHAR(4),
   LocName VARCHAR(32) NOT NULL, CONSTRAINT PK_LOCATION PRIMARY KEY
   (LocNo), CONSTRAINT FK_FACNO FOREIGN KEY (FacNo) REFERENCES FACILITY
   (FacNo) );
   ```

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.

   Answer : Null values aren't allowed for the foreign key in the *Location* table because foreign keys have a relationship to other tables. Therefore, if null values are allowed in foreign key then the 1-M relationship (as stated in problem (4)) will cause an error and the query won't be executed properly.

   The query to enforce my statements are written below :

   ```
   CREATE TABLE Location (LocNo VARCHAR(4) NOT NULL, FacNo VARCHAR(4) NOT
   NULL, LocName VARCHAR(32) NOT NULL, CONSTRAINT PK_LOCATION PRIMARY KEY
   (LocNo), CONSTRAINT FK_FACNO FOREIGN KEY (FacNo) REFERENCES FACILITY
   (FacNo) );
   ```

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.

   Answer :

   ```
   CREATE TABLE Facility (FacNo VARCHAR(4) NOT NULL, FacName VARCHAR(32)
   NOT NULL, CONSTRAINT PK_FACILITY PRIMARY KEY (FacNo), CONSTRAINT
   Unique_FacName UNIQUE(FacName));
   ```