DATABASE MANAGEMENT SYSTEMS
SKILL-UP / SKILL-BRIDGE EXERCISES

CONTENTS

1. Getting Started Exercises - Database Design

1.1 Design of Aqua Water Refilling Management System

System Description

One of the fastest and expanding businesses today is water refilling station. Water refilling station is small water system that has its own water purification facility producing a portable drinking water. The aqua water refilling system has their own water tank and equipment that intend on their business. So, we design a system in this kind of business in order to be on top and align on the fast-growing business that is demand now a day.

The purpose of water refilling management system is to overcome difficulties in manual operation in refilling station. The difficulty in manual system are one of the reasons why the efficiency in availing services of the clients is not satisfying and keeping of records is often misplaced and not secure. This system will be programmed to java that can enable the user to record things that are being purchased by the clients and it be created using SQL database.

This system manages to display the data to be filled by the user according to the information of the customer in organize manner, such that their personal details, and the services they want to avail as well as the payment on the transaction they purchased. The system keeps the information of the customer and the details of what they purchased.

The system coordinates the arrangement on delivery of products. It consists all the records for the location of the clients, date of transaction, schedule of delivery, contact number and the person assign to deliver and the payment of customer to the quantity of product that about to deliver. The system also views the information about the availability of the products as well as the containers. The system views the available containers to provide stocks again.

This system also manages the information of the employees that a refilling station must have just like front liner, cashier, technical assistant, and delivery an. It stores the information in organize so that it easy to the owner to access on the detail of his/her employee. Upon having this system, it will provide the capacity to the owner and clients to transact without spending time and effort.

Basic features:

- Manage user (add, list updates)
- Delivery (confirm, cancel)
- Products (add, delete, update).

1.2 Entity Relationship Diagram

Figure 1 Proposed water refilling management system Entity Relationship Diagram shows the system entity relationship in each entity and their supposed function in each relationship.

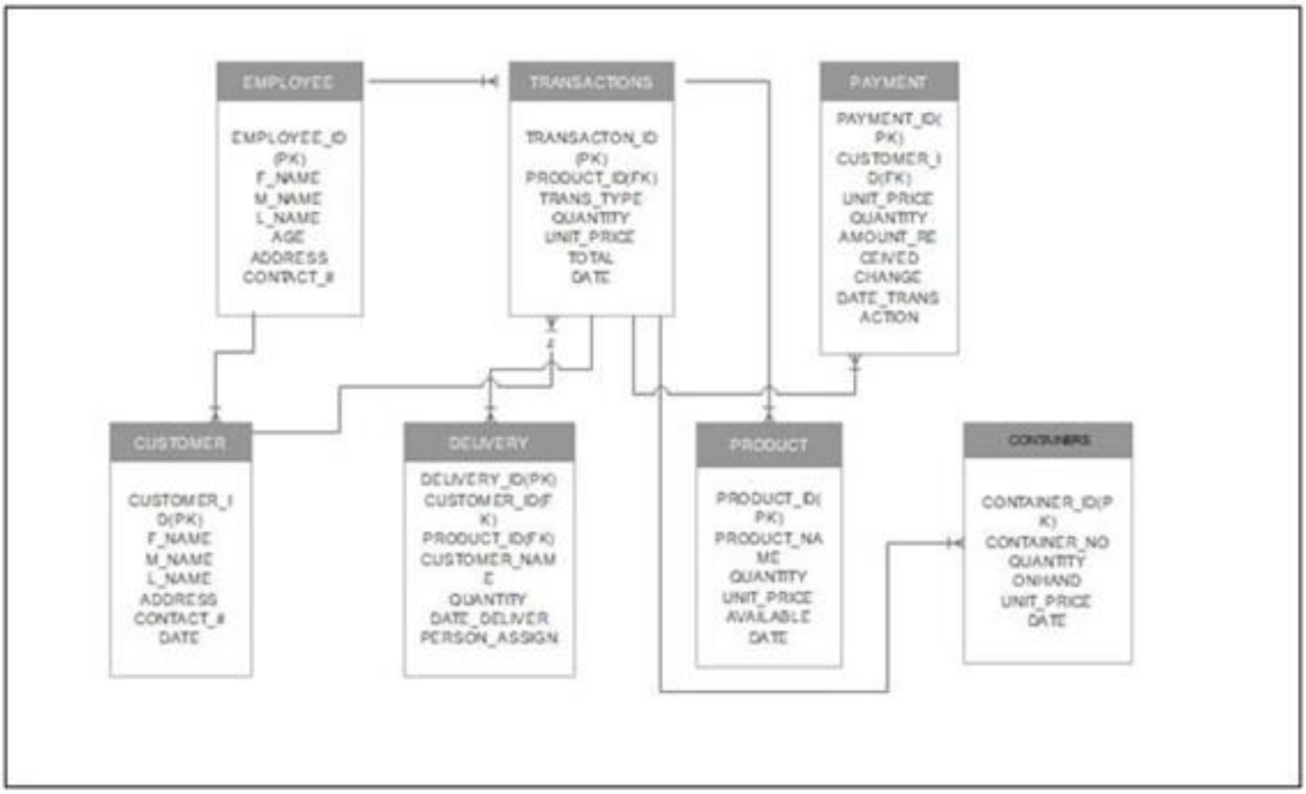


Figure 1.Aqua Water Refilling Management System Entity Relationship Diagram

Based on figure 1, the proposed aqua water refilling management system entity relationship diagram are the entity of the proposed water refilling management system database, which are represented by tables, the tables are made to meet the required specification of the system and provide a much specific details of the each entities within the system.

1.3 Data Dictionaries

The table below provides the entire database tables details such as field names, descriptions, data type and character length.

Table 1. Employee

FIELD NAMES	DESCRIPTION	TYPE	LENGTH
EMPLOYEE_ID(pk)	EMPLOYEE ID NUMBER	INT	11
F_NAME	EMPLOYEE FIRST NAME	VARCHAR	50
M_NAME	EMPLOYEE MIDDLE NAME	VARCHAR	50
L_NAME	EMPLOYEE LAST NAME	VARCHAR	50
AGE	AGE OF EMPLOYEE	VARCHAR	11
ADDRESS	LOCATION OF EMPLOYEE	VARCHAR	50
CONTACT_#	NUMBER OF EMPLOYEE	VARCHAR	11

Table 2. Customer

FIELD NAMES	DESCRIPTION	TYPE	LENGTH
CUSTOMER_ID(PK)	CUSTOMER ID NUMBER	INT	11
F_NAME	CUSTOMER FIRST NAME	VARCHAR	50
M_NAME	CUSTOMER MIDDLE NAME	VARCHAR	50
L_NAME	CUSTOMER LAST NAME	VARCHAR	50
ADDRESS	LOCATION OF CUSTOMER	VARCHAR	50
CONTACT_#	NUMBER OF CUSTOMER	VARCHAR	11
DATE	DATE OF TRANSACTION	DATE TIME	11

Table 3. Product

FIELD NAMES	DESCRIPTION	TYPE	LENGTH
PRODUCT_ID(PK)	PRODUCT ID NUMBER	INT	11
PRODUCT_NAME	PRODUCT NAME	VARCHAR	50
QUANTITY	QUANTITY OF PRODUCTS	INT	11
UNIT_PRICE	PRICE OF PRODUCTS	VARCHAR	50
DATE	DATE	DATE	11

Table 4. Delivery

FIELD NAMES	DESCRIPTION	TYPE	LENGTH
DELIVERY_ID(PK)	DEELIVERY ID NUMBER	INT	11
CUSTOMER_ID(FK)	CUSTOMER ID NUMBER	INT	11
PRODUCT_ID(FK)	PRODUCT ID NUMBER	INT	11
TRANSACTION_ID	TRANSACTION ID NUMBER	INT	11
QUANTITY	QUANTITY OF PRODUCTS	INT	11
DELIVERY_MAN	DELIVERY MAN	VARCHAR	50
DATE_DELIVER	DATE TO DELIVER	DATE TIME	11

Table 5. Containers

FIELD NAMES	DESCRIPTION	TYPE	LENGTH
CONTAINER_ID	CONTAINER ID NUMBER	INT	11
CONTAINER_NO	CONTROL NUMBER	INT	50
QUANTITY	QUANTITY ORDER OF CUSTOMER	INT	20
ON_HAND	AVAILABLE CONTAINERS	VARCHAR	50
UNIT_PRICE	PRICE OF CONTAINERS	VARCHAR	11

Table 6. Payment

FIELD NAMES	DESCRIPTION	TYPE	LENGTH
PAYMENT_ID	PRODUCT ID NUMBER	INT	11
CUSTOMER_ID	CUSTOMER ID NUMBER	INT	50
TRANSACTION_ID	DATE OF TRANSACTION	DATE TIME	11
TOTAL	TOTAL OFPURCHASED PRODUCT	INT	11
AMOUNT_RECEIVED	AMOUNT RECEIVED	INT	11
CHANGED	CHANGED	INT	11

Table 7.Transaction

FIELD NAMES	DESCRIPTION	TYPE	LENGTH
TRANSACTION_ID	PRODUCT ID NUMBER	INT	11
PRODUCT_ID	PRODUCT ID NUMBER	INT	50
TRANS_TYPE	TYPE OF TRANSACTION	VARCHAR	11
UNIT_PRICE	PRICE OF PRODUCT	VARCHAR	50
QUANTITY	QUANTITY OF PRODUCT	INT	11
TOTAL	TOTAL COST OF TRASACTION	INT	11
DATE	DATE	DATE	11

1.4 Select Statement Using Different Functions

1. Create a query that displays the complete names of all customer.
2. Create a query that displays all the name of products.
3. Show all the total number of each container.
4. Create a query that displays the sum of the ALKALINE products.
5. Show all the list of customer who transact in the month February 28.
6. Write a query that display the maximum amount received.
7. Write a query that displays the list of customer transaction details that have balance more than 100.
8. For each transaction display the transaction type and calculate the total number of customer choose the transaction.
9. Show all the number of customer who transact in each address.
10. Display each transaction type, date of transaction, and date of the week on which the transaction done. Label the column DAY. Order the results by the day of the week.
11. Write a query that display product name in lowercase and total number of each product.
12. Write a query that displays the total number of transaction.
13. Create a query that display the date and amount received.
14. Show all employee full names and contact numbers.
15. Show the maximum products.
16. For each transaction, display the transaction number, product id, unit price, and total amount to be paid discounted by 1% and expressed as a whole number.
17. Create a query that displays the average of amount tender by the customer.
18. Write a query that display the customer last names starting with M, R, T, and it’s length.
19. Display customer last name, first name and date of transaction who transact January 01, 2019 up to March 28, 2019.
20. Display the total of amount received.

1.5 Select Statements using Sub Query

1. Write a query that display the product name, price and the product that has many number of availability.
2. Write a query that displays the customer last name, first name who lived in Hyderabad.
3. Write a query that displays the last name contact number of employees who lived in Hyderabad.

1.6 Select Statements using Count and Group Functions

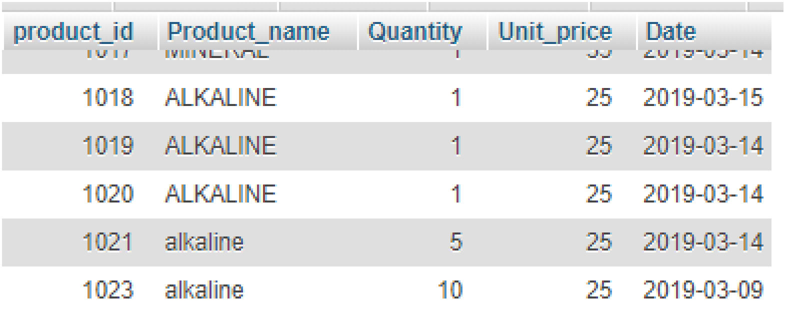
1. Display the name and the total number of containers.
2. Create a query that displays the number of customer that transact on the address in the database.

1.7 Select Statement Using Different Joins

1. Create a query that display the customer full name, total cost of purchased product, the amount tender and the changed of the customer.
2. Show the list of product that is purchased through refill.
3. Show the list of customer number, last name, and the employee first name.
4. Write a query that will display the delivery id, customer complete name in upper case format. Show also the list of delivery man and the date when the product will be deliver.
5. Display the product name, transaction type and the total cost.

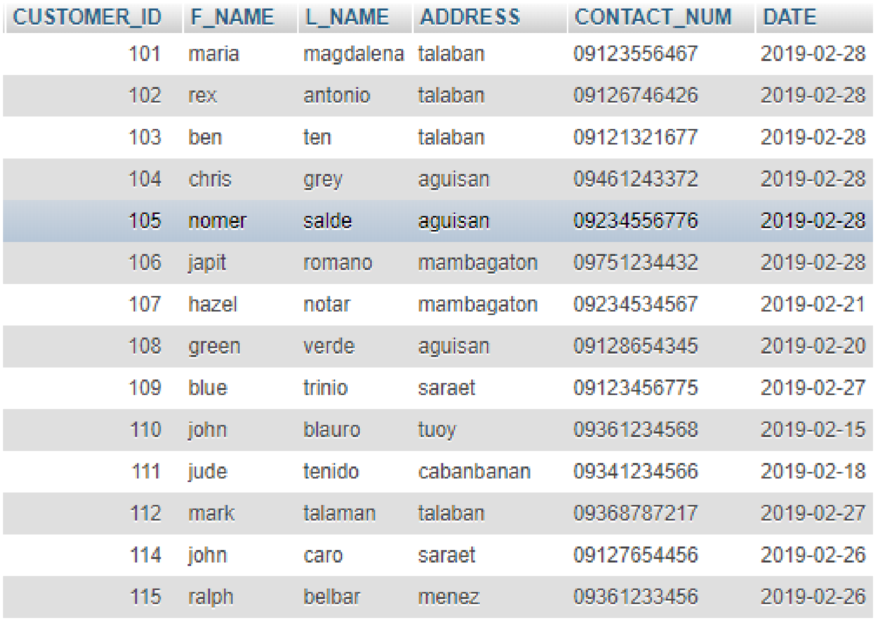
1.8 Insert Statement

Create a query to insert 10 new alkaline product and its price is 25 pesos in March 23 2019.



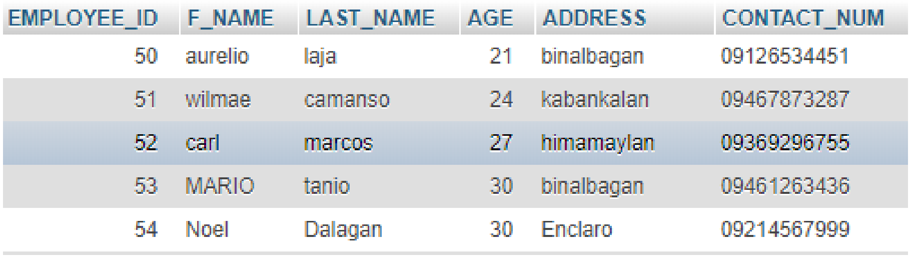
product_id	Product_name	Quantity	Unit_price	Date
1017	MINERAL	1	35	2019-03-14
1018	ALKALINE	1	25	2019-03-15
1019	ALKALINE	1	25	2019-03-14
1020	ALKALINE	1	25	2019-03-14
1021	alkaline	5	25	2019-03-14
1023	alkaline	10	25	2019-03-09

Create a query to insert new customer whose named is Jason Tenido living in Brgy.4 and has contact number of 09122345757 while Carlos Hilado lived in Brgy.Carabalan and has contactnumber of 09367281123.



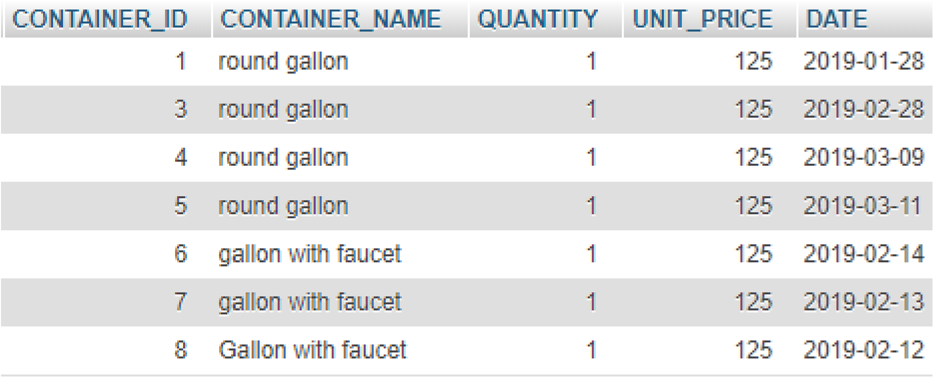
CUSTOMER_ID	F_NAME	L_NAME	ADDRESS	CONTACT_NUM	DATE
101	maria	magdalena	talaban	09123556467	2019-02-28
102	rex	antonio	talaban	09126746426	2019-02-28
103	ben	ten	talaban	09121321677	2019-02-28
104	chris	grey	aguisan	09461243372	2019-02-28
105	nomer	salde	aguisan	09234556776	2019-02-28
106	japit	romano	mambagaton	09751234432	2019-02-28
107	hazel	notar	mambagaton	09234534567	2019-02-21
108	green	verde	aguisan	09128654345	2019-02-20
109	blue	trinio	saraet	09123456775	2019-02-27
110	john	blauro	tuoy	09361234568	2019-02-15
111	jude	tenido	cabanbanan	09341234566	2019-02-18
112	mark	talaman	talaban	09368787217	2019-02-27
114	john	caro	saraet	09127654456	2019-02-26
115	ralph	belbar	menez	09361233456	2019-02-26

Create a query to insert new employee whose name is roel gecosala 27 years old who lived in bgry.enclaro and has a contact number of 09214567999.



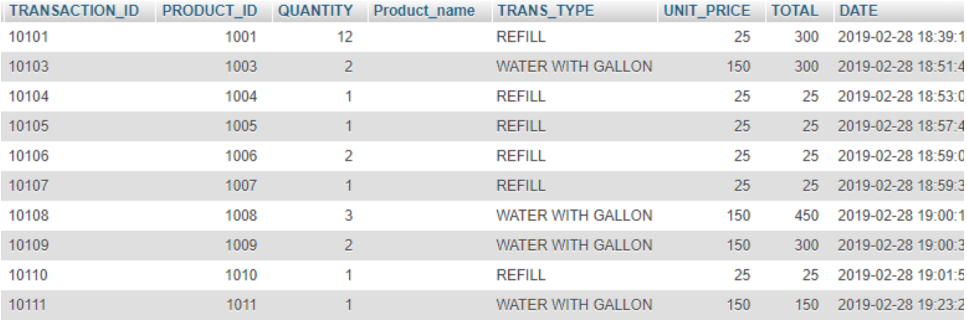
EMPLOYEE_ID	F_NAME	LAST_NAME	AGE	ADDRESS	CONTACT_NUM
50	aurelio	laja	21	binalbagan	09126534451
51	wilmae	camanso	24	kabankalan	09467873287
52	carl	marcos	27	himamaylan	09369296755
53	MARIO	tanio	30	binalbagan	09461263436
54	Noel	Dalagan	30	Enclaro	09214567999

Create a query to insert new 10 round container and cost of 125.



CONTAINER_ID	CONTAINER_NAME	QUANTITY	UNIT_PRICE	DATE
1	round gallon	1	125	2019-01-28
3	round gallon	1	125	2019-02-28
4	round gallon	1	125	2019-03-09
5	round gallon	1	125	2019-03-11
6	gallon with faucet	1	125	2019-02-14
7	gallon with faucet	1	125	2019-02-13
8	Gallon with faucet	1	125	2019-02-12

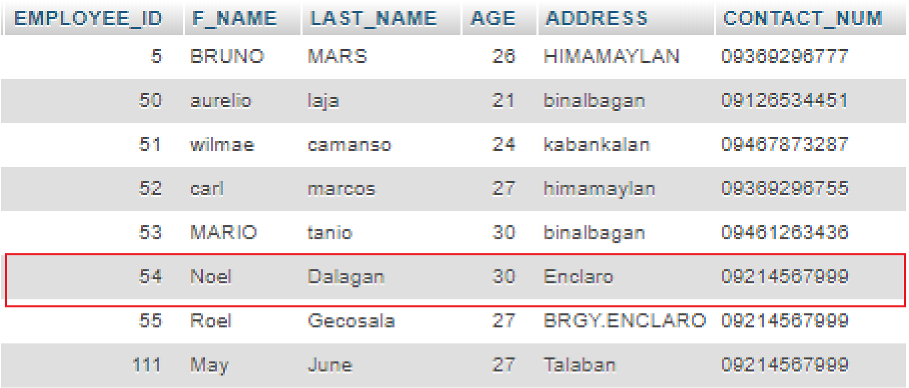
Create a query to insert new transaction such that transasction id is 10112, product id is 1012 ,quantity is 5, transaction type is refill, price is 25 and has a total cost of 125.



TRANSACTION_ID	PRODUCT_ID	QUANTITY	Product_name	TRANS_TYPE	UNIT_PRICE	TOTAL	DATE
10101	1001	12		REFILL	25	300	2019-02-28 18:39:1
10103	1003	2		WATER WITH GALLON	150	300	2019-02-28 18:51:4
10104	1004	1		REFILL	25	25	2019-02-28 18:53:0
10105	1005	1		REFILL	25	25	2019-02-28 18:57:4
10106	1006	2		REFILL	25	25	2019-02-28 18:59:0
10107	1007	1		REFILL	25	25	2019-02-28 18:59:3
10108	1008	3		WATER WITH GALLON	150	450	2019-02-28 19:00:1
10109	1009	2		WATER WITH GALLON	150	300	2019-02-28 19:00:3
10110	1010	1		REFILL	25	25	2019-02-28 19:01:5
10111	1011	1		WATER WITH GALLON	150	150	2019-02-28 19:23:2

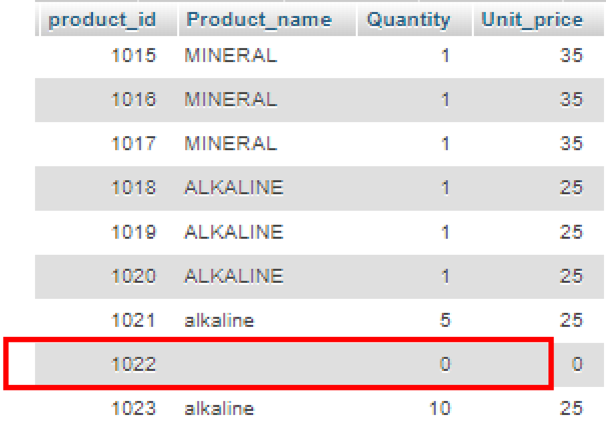
1.9 Update Statement

Create a query to change the last name of noel dalagan to delagon.



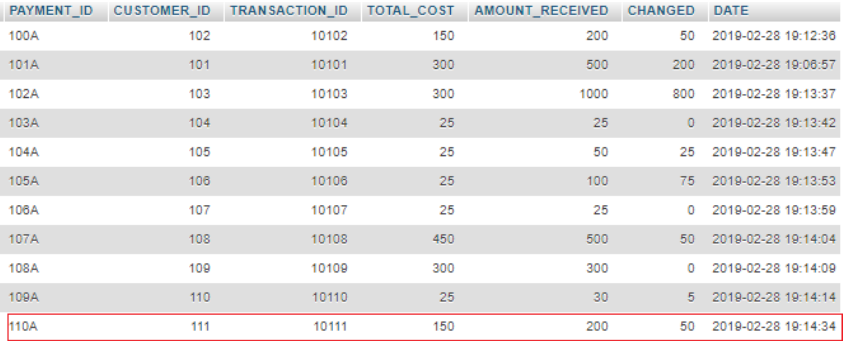
EMPLOYEE_ID	F_NAME	LAST_NAME	AGE	ADDRESS	CONTACT_NUM
5	BRUNO	MARS	26	HIMAMAYLAN	09369296777
50	aurelio	laja	21	binalbagan	09126534451
51	wilmae	camanso	24	kabankalan	09467873287
52	carl	marcos	27	himamaylan	09369296755
53	MARIO	tanio	30	binalbagan	09461263436
54	Noel	Dalagan	30	Enclaro	09214567899
55	Roel	Gecosala	27	BRGY.ENCLARO	09214567899
111	May	June	27	Talaban	09214567899

Create a query to change the information on the product id 1022 .



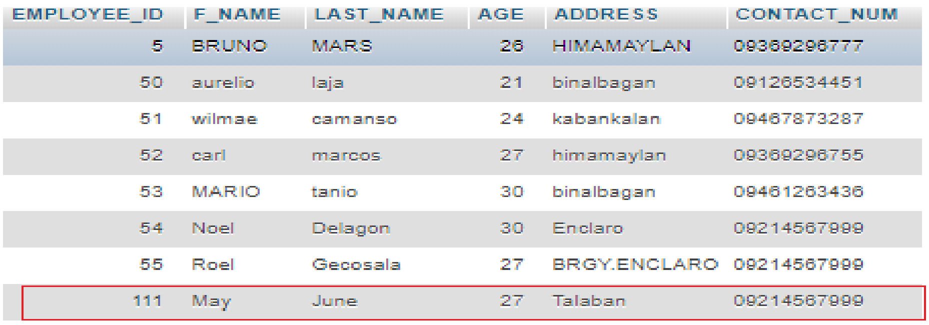
product_id	Product_name	Quantity	Unit_price
1015	MINERAL	1	35
1016	MINERAL	1	35
1017	MINERAL	1	35
1018	ALKALINE	1	25
1019	ALKALINE	1	25
1020	ALKALINE	1	25
1021	alkaline	5	25
1022		0	0
1023	alkaline	10	25

Write a query to change the amount render by the customer having the payment id of 110A.



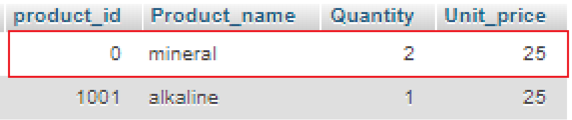
PAYMENT_ID	CUSTOMER_ID	TRANSACTION_ID	TOTAL_COST	AMOUNT_RECEIVED	CHANGED	DATE
100A	102	10102	150	200	50	2019-02-28 19:12:36
101A	101	10101	300	500	200	2019-02-28 19:06:57
102A	103	10103	300	1000	800	2019-02-28 19:13:37
103A	104	10104	25	25	0	2019-02-28 19:13:42
104A	105	10105	25	50	25	2019-02-28 19:13:47
105A	106	10106	25	100	75	2019-02-28 19:13:53
106A	107	10107	25	25	0	2019-02-28 19:13:59
107A	108	10108	450	500	50	2019-02-28 19:14:04
108A	109	10109	300	300	0	2019-02-28 19:14:09
109A	110	10110	25	30	5	2019-02-28 19:14:14
110A	111	10111	150	200	50	2019-02-28 19:14:34

Create a query to change the contact number of the employee named May and add suffix on her name.



EMPLOYEE_ID	F_NAME	LAST_NAME	AGE	ADDRESS	CONTACT_NUM
5	BRUNO	MARS	26	HIMAMAYLAN	09369296777
50	aurelio	laja	21	binalbagan	09126534451
51	wilmae	camanso	24	kabankalan	09467873287
52	carl	marcos	27	himamaylan	09369296755
53	MARIO	tanio	30	binalbagan	09461263436
54	Noel	Delagon	30	Enclaro	09214567999
55	Roel	Gecosala	27	BRGY.ENCLARO	09214567999
111	May	June	27	Talaban	09214567999

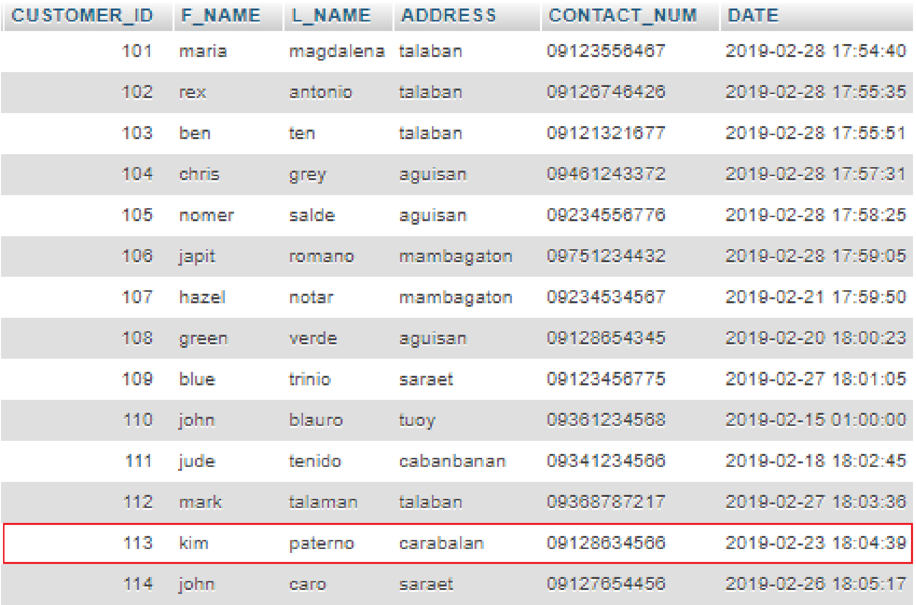
Create a query to change the product id and the price of the newly inserted product .



product_id	Product_name	Quantity	Unit_price
0	mineral	2	25
1001	alkaline	1	25

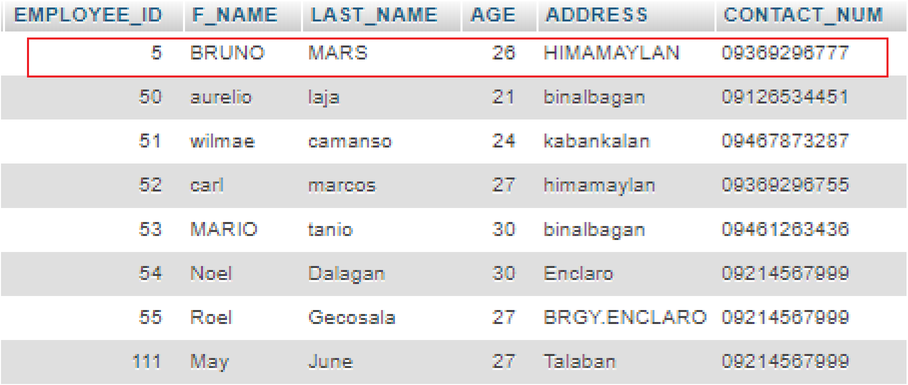
1.10 Delete Statement

Create a query to delete one data in customer record who transact on February 23, 2019.



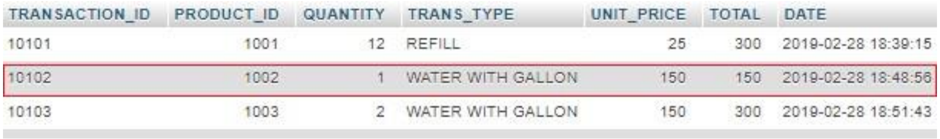
CUSTOMER_ID	F_NAME	L_NAME	ADDRESS	CONTACT_NUM	DATE
101	maria	magdalena	talaban	09123556467	2019-02-28 17:54:40
102	rex	antonio	talaban	09126746426	2019-02-28 17:55:35
103	ben	ten	talaban	09121321677	2019-02-28 17:56:51
104	chris	grey	aguisan	09461243372	2019-02-28 17:57:31
105	nomer	salde	aguisan	09234556776	2019-02-28 17:58:25
106	japit	romano	mambagaton	09751234432	2019-02-28 17:59:05
107	hazel	notar	mambagaton	09234534567	2019-02-21 17:59:50
108	green	verde	aguisan	09128654345	2019-02-20 18:00:23
109	blue	trinio	saraet	09123456775	2019-02-27 18:01:05
110	john	blauro	tuoy	09361234568	2019-02-15 01:00:00
111	jude	tenido	cabanbanan	09341234566	2019-02-18 18:02:45
112	mark	talaman	talaban	09368787217	2019-02-27 18:03:36
113	kim	paterno	carabalan	09128634566	2019-02-23 18:04:39
114	john	caro	saraet	09127654456	2019-02-26 18:05:17

Create a query to delete one employee.



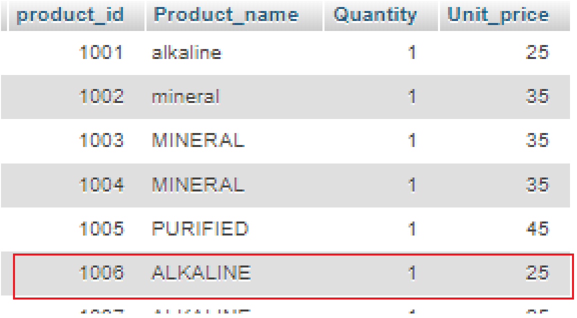
EMPLOYEE_ID	F_NAME	LAST_NAME	AGE	ADDRESS	CONTACT_NUM
5	BRUNO	MARS	26	HIMAMAYLAN	09369296777
50	aurelio	laja	21	binalbagan	09126534451
51	wilmae	camanso	24	kabankalan	09467873287
52	carl	marcos	27	himamaylan	09369296755
53	MARIO	tanio	30	binalbagan	09461263436
54	Noel	Dalagan	30	Enclaro	09214567999
55	Roel	Gecosala	27	BRGY.ENCLARO	09214567999
111	May	June	27	Talaban	09214567999

Write a query that will delete a one data on transaction.



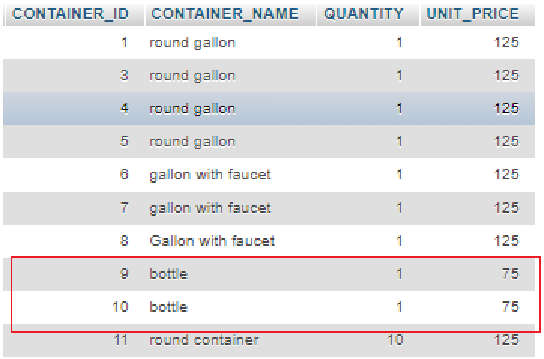
TRANSACTION_ID	PRODUCT_ID	QUANTITY	TRANS_TYPE	UNIT_PRICE	TOTAL	DATE
10101	1001	12	REFILL	25	300	2019-02-28 18:39:15
10102	1002	1	WATER WITH GALLON	150	150	2019-02-28 18:48:56
10103	1003	2	WATER WITH GALLON	150	300	2019-02-28 18:51:43

Write a query that will remove the product has the product id of 1006.



product_id	Product_name	Quantity	Unit_price
1001	alkaline	1	25
1002	mineral	1	35
1003	MINERAL	1	35
1004	MINERAL	1	35
1005	PURIFIED	1	45
1006	ALKALINE	1	25
1007	ALKALINE	1	25

Remove the container named bottle .



CONTAINER_ID	CONTAINER_NAME	QUANTITY	UNIT_PRICE
1	round gallon	1	125
3	round gallon	1	125
4	round gallon	1	125
5	round gallon	1	125
6	gallon with faucet	1	125
7	gallon with faucet	1	125
8	Gallon with faucet	1	125
9	bottle	1	75
10	bottle	1	75
11	round container	10	125

1.11 Sample Records

1. Write a query to show all the records of customers.
2. Write a query to show all the records of containers.
3. Write a query to show all the records of product.
4. Write a query to show all the records of employee.
5. Write a query to show all the records of transaction.

6. Write a query to show all the records of payment.
7. Write a query to show all the records of delivery.

2. DESIGN OF CLINIC MANAGEMENT SYSTEM

2.1 Introduction

The clinic management system is based on the idea of providing an automated system. It saves time and extra overhead to perform the action. It will contain five main modules:

- Admin
- Doctor
- Patient
- Staff
- Visitor.

Admin will be the system administrator who keeps the record of the clinic intact.
He can personalize the users' profiles. Visitors are the unregistered users in the system, who can view hospital details, how to contact them etc. Doctors, patients, and staff will be the registered users in the system.

Details: Here are some main functionalities of each type of users involved in the system:

Admin

- It is the main entity of the system. Admin will the system administrator of the system. He /she will have the power to manage the users involved in the system.
- Those people can be a patient, a doctor, a nurse, or a staff. He/she can insert any user into the system.
- It might be possible if any doctor or any other staff leaves the clinic. So that doctor or staff is irrelevant to that clinic. He/she can delete that user from the system after some period of time.
- Consider the scenario when any staff gets the promotion.
- Now, that information needs to be reflected in the system as well. The admin can update it whenever he wants. There will be only one admin in the system.

Doctor

Registration:

The doctor will have to register himself/herself into the system first. To register, he will have to enter some personal information.
He will be asked to input his full name, password he wants to use for login into the system, re-enter password (so that he distinctly remembers his password) contact number, and email address.
After successful completion of the registration process, a unique id will be generated and will be displayed on the user's console. This id will be used as a login id to login into the system.

Login: If a doctor is registered into the system, with the help of his personal unique id and his personalized password, he can log in to the system and can get started.

Editing Profile:

It is an important task; the doctor has to complete before proceeding further. He/she will have to enter his/her designation in the system.

A patient cannot find a suitable doctor to help him/her in the hour of the need if a patient does not know which doctor he should consult at the right time.

The other thing the doctor has to insert is his / her address, where he/she live. If the doctor wants to change his email address or his mobile number, which he/she provided at the time of registration, he can change it any time he/she wants.

Update the Time Table

Initially, there would be a static timetable for the doctor's activities. According to his / her timetable, he will take the patient, perform the operation, etc.

Sometimes it happens that an emergency case comes to his / her attention. Now, he/she has to take it. So, he/she can reflect this activity on his/her timetable.

It will help the admin to keep a clean record of the hospital. It can be useful for several purposes.

Approve Appointment

When a patient books an appointment with the doctor, the doctor can approve the appointment, whether he can attend the patient or not.

The doctor will always approve the request for the appointment, until and unless he/she has a valid reason not to.

He/she will have to provide a valid reason if he /she cannot have an appointment with the patient. If possible, he /she will have to provide any proof.

Cancel Appointment

- What happens if the doctor is not available on the day of the appointment due to some genuine reason? And the patient arrives at the hospital.
- How that is going to benefit the patient? So, the doctor will cancel the appointment if he thinks he cannot make it on that day. But wait, this is not going to reflect on the patient panel at once.
- This status is going to be reflected on the admin side. The admin will check if another doctor is free on that day who can attend that patient. If so, an appointment for an available doctor will be booked and reflect on his interface.
- He will get the mail or SMS for this appointment. If not, the appointment will be canceled. Though, that would be a rare

Patient

Registration

In order to get benefits from the system, the person needs to register himself/herself as a patient in the system. He/she needs to provide his / her name, date of birth, email address, and password at the time of registration. After successful registration, he /she will be given a unique id. That id will be used to login into the system.

Login

Only a registered patient can log in to the system.

He/she will have to enter a unique id and password.

If both id and password match with the system, he /she will be granted access to the patient panel.

2.2 Make an appointment with the doctor

- This is the main feature of the system. The patient is involved in the system only because he needs medical assistance.
- A person cannot do productive work if he/she is not feeling well. He/ she can go to the appointment section in his / her interface.
- There is some information that needs to be filled in in order to make an appointment. Such as: what are the symptoms? since when he/she is having this problem? Etc.
- After filling in the information, a unique appointment id will be given to the patient and he/she can provide this id at the hospital counter to consult with the doctor.

Payment

- In order to complete the appointment process, the patient needs to make the payment.
- The charges will be shown at the interface.
- The patient can choose whether he/she will make the payment as cash at the hospital or he/she can choose the online method to make the payment.
- The online method will be secure so that his/her debit/credit card information cannot be misused.

Cancel Appointment

- Due to some reason, a patient, who has made an appointment, cannot attend the doctor that day.
- The reason can be personal or he might have got well.
- He/she just has to enter the appointment id.

Home Remedies

- This feature will be available to the registered patients of the system.
- Sometimes, a person who does not well does not need a doctor. Yes, it happens.
- Now, you ask how? I will tell you. Suppose a person's head is aching.
- Does he/she need a doctor? Maybe yes or maybe no. At first, he/she can check out if it can be controlled.
- So, he/she can check out the "Home Remedies" section and apply the formula for the headache.
- If it works then that's fine, if not he/she can make an appointment with the doctor.
- There would be a whole list of remedies that will be easily available to the patients.

Health Tip of the Day

- Yes, this is another attractive feature that is available to the registered patients in the system.
- Every day a new tip will be notified to the users.
- He/she can implement those tips in their life and can make it happy.
- The main purpose of this section is to make each person a healthy life.

Staff

This is the fourth module of the system. Other than doctor and patient, there are several people involved in the system such as a nurse, compounder, cashier, receptionist, etc.

Registration – They must register themselves into the system. This section is the same as doctor and patient. They also have to provide their personal information.

Login – When a staff is registered into the system, he/she can log in to the system by providing the unique id and password.

Editing Profile – A staff needs to complete his / her profile. He/she has to provide his / her designation, address, etc.

Apply for Leave – A staff can also apply for leave. This leave request should contain a detailed reason. The system administrator will approve the leave only if he finds the reason legit, otherwise can reject it. The staff can check the leave status in the leave section.

Change Password – This is the common feature for the previous three modules. A user can change his / her password. He/she just has to provide the old password and has to write the new password.

Forgot Password – This is another common feature for the users. If a user forgets his password, he just has to answer some security questions which were answered by him/her at the time of registration.

Log out – A logged-in user can always log out of the system when his work is done, just by hitting the log-out. It is available to all registered users.

2.3 Visitors

This will be the last module for the unregistered user in the system. The visitor can check for the doctor list, how qualified doctors are into the system, what main facilities this clinic system has such as X-ray availability, CT-scan availability, blood test availability, etc. If the visitor is satisfied with the service this system has, he/she can register himself/herself and can get benefit from it.

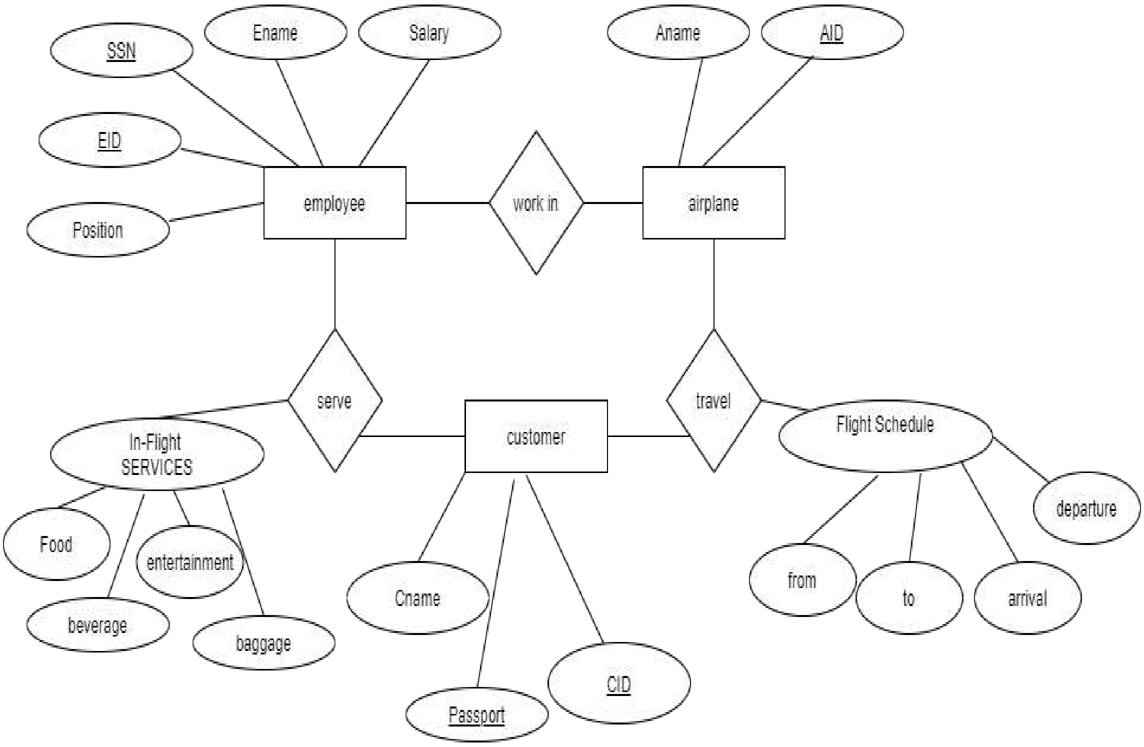
3. DESIGN OF FLIGHT DATABASE SYSTEM

3.1 Objectives

To create a flight database systems

- Employees work in Airplanes, many employees work in many airplanes
- Customers board airplanes, many customers can board many airplanes
- Employees serve customers, many employees serve many customers
- Customers can avail to services like food beverage and entertainment

3.2 ER Diagram



3.3 Relational schema

- Airplane (Aname, AID)
- Travels (Departure, Arrival, To, From,AID,CID) AID is foreign key referencing airplane, CID is foreign key referencing customer
- Serves (Food, Beverage, Entertainment, Baggage, EID,CID) CID is foreign key referencing customer, EID is foreign key referencing employee
- Employee (EID, SSN, ENAME, Position, Salary)
- Works(AID,EID) AID is foreign key referencing airplane, EID is foreign key referencing employee
- Customer(Cname,Passport,CID)

3.4 BCNF schema

- Airplane (Aname, AID)
- AID is the Candidate Key
- AID→(Aname)
- Travels (Departure, Arrival, To, From,AID,CID) AID is foreign key referencing airplane, CID is foreign key referencing customer
- Serves (Food, Beverage, Entertainment, Baggage, EID,CID) CID is foreign key referencing customer, EID is foreign key referencing employee
- Employee (EID, SSN, ENAME, Position, salary)
- EID is the candidate key
- EID→ (Ename, SSN, Position,Salary)
- Works(AID,EID) AID is foreign key referencing airplane, EID is foreign key referencing employee
- Customer(Cname,Passport,CID)
- CID is the candidate key
- CID(Cname,Passport)

Schema is in BCNF

3.5 SQL

Create table airplane (AID varchar (20) PRIMARY KEY, aname varchar (50) not null);

Create table employee (EID varchar (20) PRIMARY KEY, ename varchar (50) not null, SSN int(30) unique not null, position char(30) not null, salary char(30) not null);

Create table customer (CID varchar (20) PRIMARY KEY, cname varchar (50) not null, passport varchar (30) unique not null);

create table works(AID varchar(20), EID varchar (20), FOREIGN KEY (AID) REFERENCES airplane (AID) on delete set null, FOREIGN KEY (EID) REFERENCES employee (EID) on delete set null);

create table travels(arrival char(30) not null, departure char(40) not null, too char(30) not null, fro char(30) not null, AID varchar(20), CID varchar(20), FOREIGN KEY (AID) REFERENCES airplane (AID) on delete set null, FOREIGN KEY (CID) REFERENCES customer (CID) on delete set null);

```
create table serves( food char(30), beverage char(30), entertainment char(30), baggage char(30), AID varchar(20), CID varchar(20), FOREIGN KEY (AID)REFERENCES airplane (AID) on delete set null, FOREIGN KEY (CID) REFERENCES customer (CID) on delete set null.
```

View and triggers

```
create view flight_schedule as
select arrival, departure, too,fro, aname from
(travels natural join airplane)group by aname;
create view inflight_services as
select cname,food, beverage, entertainment, baggage
from (serves natural join customer)group by cname;
Create Trigger Excess_baggage Before Insert On inflight_services For Each Row
Begin
if(baggage >= ‘15’) then
RAISE_APPLICATION_ERROR( - 20343, 'The baggage should be <=15kg'); End if;
End;
Create Trigger Excess_baggage Before Update On inflight_services For Each Row
Begin
if(baggage >= '15') then
RAISE_APPLICATION_ERROR( - 24711, 'baggage should be <=15kg'); end if;
End;
```

INDEXING

From the airplane and employee tables, the customer table and the employee table would need an index for the reasons as mentioned below:

An index that can be used for the convenience of the employees is inflight services it could be used to keep track of which customers have ordered food and beverages. In order to provide them with prompt service for this we can make use of a secondary index as several customers may have availed the same services.

CREATE INDEX

```
serv_info ON serves(food,beverage);
```

Another index that could be required is an index of aircrafts for this we shall make use of Primary indexes where the index would be in accordance with the airplane id.

CREATE UNIQUE INDEX

```
aircraft_info ON airplane(AID);
```

In an airline the employee databases are smaller than the customer. Hence for this we shall use a primary index where the index would be arranged in accordance with the employee IDs, Thus making it easy to find employee details

```
CREATE UNIQUE INDEX emp_info ON employee(EID);
```

4. DESIGN A DATABASE SCHEMA FOR RECRUITMENT

4.1 Introduction

After we have described DB design basics in the first part of the article, let us now create a database schema for recruitment.

First of all, we need to define which information is important for company employees who search for job applicants:

For an hr-manager:

1. Companies where an applicant used to work at.
2. Positions that an applicant used to hold at these companies.
3. Skills that an applicant used at work, length of their employment at each of the companies and in each position, length of each skill use.

For a technical specialist:

1. Positions that an applicant used to hold at previous places of work.
2. Skills that an applicant used at work.
3. Projects that an applicant took part in. In addition to that, it is important to know the applicant's length of employment in each position and in each project as well as the length of each skill use.

Let us first identify the necessary entities:

- Employee
- Company
- Position
- Project
- Skill

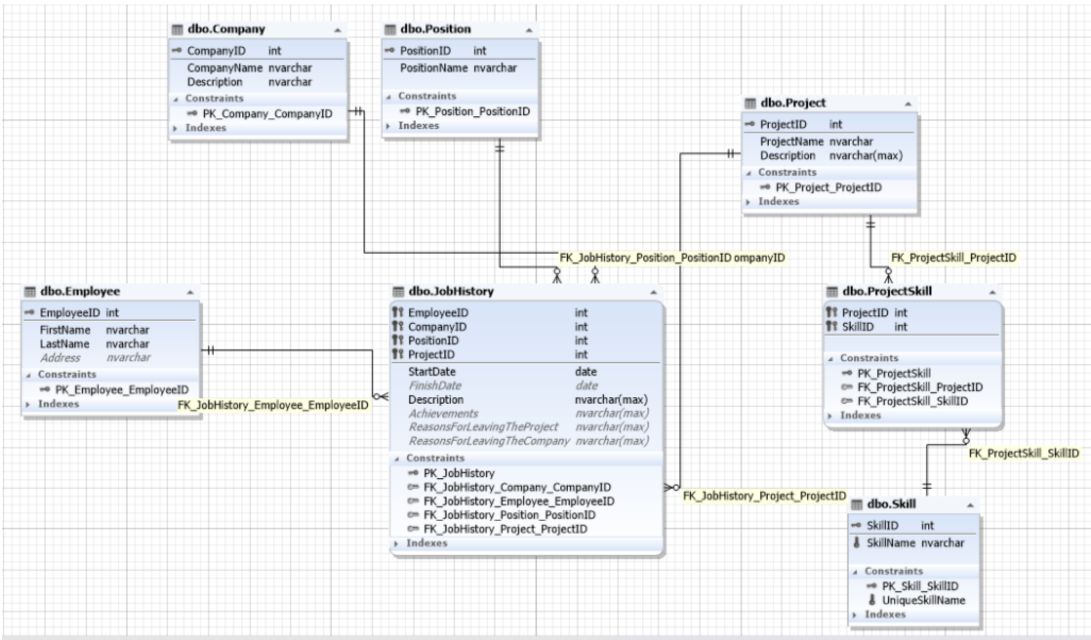
Company and employee have a many-to-many relationship since an employee could work for different companies, and the companies have many employees.

The same works for position and employee, because many employees can work in one position within one company as well as in different companies. Herewith, an employee could work in different positions both within one company and at many different ones. As a result, the relationship between position and company is many-to-many as well.

The "project" entity follows the same logic: the project relates to all other above-mentioned entities as many-to-many.

For simplicity, let us say that an employee uses one set of skills in a project. Then, the relationship between project and skill is also many-to-many.

Considering the importance of specifying an employee's length of employment at this or that company, in a particular position, and in a certain project, our database schema may have the following ER diagram.



Img.11. The database schema for a recruitment service

Here, the JobHistory table represents the entity of each employee’s job history, that is, the very resume that implements the many-to-many relationship between employee, company, positions, and project.

Project and skill relate as many-to-many, thus, they are linked with the help of the ProjectSkill entity.

If you understand the relationship between subjects, and subjects and objects, meaning database design norms, you can create a similar schema on a piece of paper in under an hour.

Here, we could simplify the schema and data addition, if we put skill in the project entity through semi-structured data (NoSQL) in the form of XML, JSON, or simply list the names of skills with a semicolon. But this would make it difficult to select grouping by skills and filter by certain skills.

5. DESIGN A DATABASE SCHEMA FOR MANAGEMENT OF COMPANY RELOCATIONS

5.1 Introduction

Design the database for the management of company relocations. The database must contain a list of vans suitable for relocations. The vans are identified by their plate, and they are characterized by their model and, if known, by their volume in cubic meters. Of all the vans, some are authorized to transport special materials, and only for such vans, a list with the certifications of the special materials known to be allowed has to be stored. The database must contain a list of warehouses, identified by a code and characterized by their address and the name of the company to which they belong. You are requested to keep track of all the relocations made. The relocations are identified by the date and by the van with which they are made, and they are characterized by the name of the driver who carried it out. Each relocation is also characterized by the departure warehouse and departure time, and by the arrival warehouse and arrival time.

6. DESIGN A DATABASE SCHEMA FOR MANAGEMENT OF VAN REPAIRS

6.1 Introduction

Design the database for the management of van repairs. The database must contain a list of vans identified by their plate and characterized by the year of registration. Among the several types of vans, the capacity of the battery is known for those equipped with an electric motor. The database must also include the list of repair shops, identified by a unique code, and characterized by their address. The repair shops can be in partnership with some companies. For each partner company, its VAT number, name, and possibly the list of telephone numbers is known. Note that each company can be in partnership with multiple repair shops and you are requested to keep track only of those companies having a partnership. You are also requested to keep track of the repairs carried out over time on each van. The date, cost and duration in hours are known for each repair. Note that a van can undergo multiple repairs on the same day but in different repair shops. A repair shop can carry out at most one repair for the same van on the same day.

7. DESIGN A DATABASE SCHEMA FOR MANAGEMENT OF RENTAL CONTACTS

7.1 Introduction

Design the database for the management of rental contracts. The database must contain a list of properties available for rent. The properties are identified by a land code, they are characterized by the address, the area in square meters and the list of rooms. Among the different properties available, for the apartments, the floor number is known. The database must contain a list of agencies, identified by a code, and characterized by the address and the list of phone numbers. You are requested to keep track of all the rental contracts of the agencies for each property. Different rental contracts may have been stipulated for the same property, with the same agency or with different agencies. A rental contract is characterized by a start date and an end date, by a monthly price, and can be renewable or non-renewable. A property can only have one rental contract for the same time period.

A contract is associated with a single property and a single agency.

8. Design a database schema for Management of real estate agent mandate

8.1 Introduction

Design the database for the management of agent mandates at some real estate agencies. The database must contain a list of agents. Agents are identified by the Tax ID, they are characterized by their full name, the telephone number, and they may have an e-mail address. Each agent covers one or more geographical areas.

Every geographical area is identified by the main city and it is characterized by the list of possible other municipalities included in the area, by the total number of inhabitants, and by the territorial area in square kilometers. The database must contain a list of real estate agencies. Real estate agencies are identified by an unique identifier, and they are characterized by an address. Agencies can be either independent or franchised. For what it concerns independent agencies, the Tax ID of the business owner is known. Regarding the franchised agencies, only the name of the retail chain which they belong to is known. You are requested to keep track of all the mandates that agencies gave to their agents. An agent can be given more mandates from the same agency at separate times. An agency may have given more mandates to the same agent at different times. An agent can simultaneously have mandates from different agencies. You are requested to keep track of the start date, the end date, and the type of each mandate.

9. Design a database schema for Management of staff shifts in company buildings

9.1 Introduction

Design the database for the management of staff shifts in different company buildings. The database must contain a list of employees. Employees are identified by a unique code within the company, they are characterized by their name, email address, and the list of qualifications.

Employees can be either factory workers or clerks; for clerks their role is known. The database must contain the list of the company buildings, which are identified by a unique code and characterized by their address and the list of telephone numbers. You are requested to keep track of all the work shifts of each employee at the various buildings. A work shift is characterized by a date, a start time and an end time, the building where it takes place, and it can be either ordinary or overtime. A shift is associated with only one employee and one building. The same employee can have work shifts in different buildings; however, an employee can work only one shift on each day.

10. Design a database schema for Management of maintenance inspections

10.1 Introduction

Design the database for the management of maintenance inspections executed by workers at different machinery. The database must contain a list of workers. The workers are identified by a unique code and are characterized by their name, surname, and may have a telephone number. Each worker is assigned to one or more products. Each product is identified by the bar code and is characterized by the list of categories to which it belongs, the duration in hours of its production process and the cost. The database must contain a list of machinery. Each machinery is identified by a unique code and is characterized by a brand and a model. The machinery can be either production machinery or quality control machinery. For production machinery, the maximum energy consumption is known. The quality parameter is known for quality control machinery. You are requested to keep track of all the maintenance inspections that workers have carried out on the various machinery. A worker can carry out multiple inspections on the same machinery at different times. Machinery can receive multiple inspections from the same worker at different times. At the same time, a worker can have multiple inspections on different machinery in progress. You are requested to keep track of the start date and time, and the end date and time of each inspection.

11. Exercise on Database Design using E-R Model

11.1 Introduction

Construct an E-R diagram for insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars and has one or more premium payments associated with it. Each payment is for a particular period of time, and has an associated due date, and the date when the payment was received.

Consider a database that includes the entity sets student, course, and section from the university schema and that additionally records the marks that students receive in different exams of different sections.

- Construct an E-R diagram that models exams as entities and uses a ternary relationship as part of the design.
- Construct an alternative E-R diagram that uses only a binary relationship between student and section. Make sure that only one relationship exists between a particular student and section pair, yet you an represent the marks that a student gets in different exams.

Design an E-R diagram for keeping track of the scoring statistics of your favorite sports team. You should store the matches played, the scores in each match, the players in each match, and individual player scoring statistics for each match.

Consider an E-R diagram in which the same entity set appears several times, with its attributes repeated in more than one occurrence. Why is allowing this redundancy a bad practice that one should avoid?

A university DB contains information about professors (identified by SIN) and courses (identified by course ID). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it, and list all candidate keys of the Teaches relationship set.

a. Professors can teach the same course in several semesters, and each offering must be recorded.

b. Professors can teach the same course in several semesters, but only the most recent such offering needs to be records.

Assume the above Situation (b) applies in all subsequent situations. Then, draw an ER diagram that describes each of the following situations:

- Every professor teaches a course, and every course is taught by some professor.
- Every professor teaches exactly one course, and every course is taught by exactly one professor

Consider the following company DB. It stores information about employees (identified by SIN, with s alary and phone attributes) and departments (identified by dept ID, with department name and budget as attributes). Employees work in departments.

The DB records the interval during which an employee works for a department.

Draw an ER diagram that describes each of the following situations.

- Employee is not allowed to work in a department for two or more intervals.
- Employee is allowed to work in a department for one or more intervals.

Exercise

Create database named LIBRARY

Give names to all the constraints (except NULL constraints). The properties of the tables are defined as follows:

MEMEBERS table

CardNo - 5 characters, primary key,

Surname - up to 15 characters,
Name – as above,
Address –up to 150 characters,
Name, Surname, Birthday_date – not null,
Gender - 1 char: M or F letter,
Phone_No – up to 15 characters.

Employees table

emp_id - primery key with identity set (seed = 1, increment=1) ,
Surname, Name and Birthday_date are not null,
birthday date must be earlier than date of employment (Emp_Date),

Publishers table

pub_id is a primary key with identity set (seed=1, incerement=1),
Name, City, - not null, up to 50 characters,
Phone_No - up to 15 charakters,

Books table

BookID - primary key, 5 characters,
Pub_ID - foreign key related to Publishers,
Type - charaters, must contain one of the following values: novel, historical, for kids, poems, crime story, science fiction, science
Price is a currency field (money), not null,
Title - up to 40 characters, not null,

BOOK_LOANS table
LoanID - integer with identity set (seed = 1, increment = 1), primary key,
CardNo, BookID and emp_id are foreign keys related to Members, Books and Employees,
DateOut must be earlier than DueDate,
Penalty can't contain negative values, default is set to 0 (zero),