Name: T.Sripoornima

Class: III CSE-A

Rollno:19BCS027

Data Base Management System:

Assignment-1: (Identification Of Keys)

Consider a University which runs many courses for its students on their campus. Students enroll in the University and register/sign up for one or more courses based on their subject area. The University also provides accommodation in their hostels to the students who opt for it. Hostels are spread across different buildings and in various locations of the city. Here are the details that are maintained by the University.

Entity / activity	Attributes	Description
Student	studentid	Student identification number(Id) given to identify a student when they register for a course
	studentname	Name of the student
Course	courseid	Identification number through which a course is identified
	coursename	Title given to the course
	duration	Duration for which the course is run
Course registration	studentid	Id of the student who has enrolled in the University
	courseid	Course Id to which the student has registered/enrolled
Hostel	hostelroomnumber	Hostel room number allocated to the students who have opted for accommodation
	hostelwing	Name given to identify hostel building
	address	Address of the hostel
	studentid	Id of the student staying in the particular hostel room

With the given details, the database designers want to identify **parent** table(s) and **child** table(s), and also the **candidate**, **primary** and **foreign keys** for each table. Can you help with the identification?

Solution:

Parent Table	Child Table
Student	CourseRegistration
Course	
Hostel	

Entity	Primary Key	Candidate Key	Foreign Key
Student	studentid	Studentid,studentname	Doesnot Exist
Course	courseid	courseid ,coursename	Doesnot Exist
CourseRegistration	{courseid,Studentid}	{studentid,courseid}	{courseid,studentid}
Hostel	hostelwing	hostelwing	Studentid

Assignment-2: (Normalization)

Problem Statement:

Given below is the data in an un-normalized table. Normalize it to 1NF. Identify the problems encountered when the table is in 1NF. Subsequently normalize to 2NF and 3NF, explaining the problems faced and the solution to it.

proj_no	proj_name	emp_no	emp_name	job_level	hourly_rate_in_dollars
2023	Amsterdam travel site	101	Vincent R	5	60
		102	Pauline J	4	50
		103	Charles C	3	40
2056	Real Estate Agency	101	Vincent R	5	60
		107	David R	4	50

Solution:

Step-1: 1 NF

Proj_no	Proj_name	Emp_no	Emp_name	Job_level	Hourly_rate_in_dollars
2023	Amsterdam travel site	101	Vincent R	5	60
2023	Amsterdam travel site	102	Pauline J	4	50
2023	Amsterdam travel site	103	Charles C	3	40
2056	Real Estate Agency	101	Vincent R	5	60
2056	Real Estate Agency	107	David R	4	50

Step-2: 2 NF

Employee_Project Table,

Proj_no	Emp_no
2023	101
2023	102
2023	103
2056	101
2056	107

Employee Table,

Emp_no	Emp_name	Job_level	Hourly_rate_in_dollars
101	Vincent R	5	60
102	Pauline J	4	50
103	Charles C	3	40
107	David R	4	50

Project Table,

Proj_no	Proj_name
2023	Amsterdam travel site
2056	Real Estate Agency

Step-3: 3 NF

Employee_Project Table,

Proj_no	Emp_no
2023	101
2023	102
2023	103
2056	101
2056	107

Employee Table,

Emp_no	Emp_name	Job_level
101	Vincent R	5
102	Pauline J	4
103	Charles C	3
107	David R	4

Job Table,

Job_level	Hourly_rate_in_dollars
5	60
4	50
3	40
4	50

Project Table,

Proj_no	Proj_name
2023	Amsterdam travel site
2056	Real Estate Agency

These tables are in 3rd Normal Form. Hence, they are in normalized form.

Assignment-3: (Joins, Sub-queries)

Consider a website "Greets.com" which sells greeting cards for various occasions like birthdays, anniversaries etc. The various occasions are stored as different categories in the table greeting card. Customers can register/signup on this website to purchase and send greeting cards on various occasions online. Here are some assumptions with respect to this scenario.

- One customer can buy many cards
- Greeting cards of the same category can be purchased by the same or different customers
- More than one card can have the same price

The database design used to represent this scenario is provided below.

Table name	Fields	Description
customer	customerid	Unique identification number given to a customer
	custname	Name of the customer
greetingcard	cardid	Unique identification number of a greeting card
	customerid	Id of customer who has purchased a card. This refers to the customerid from customer table.
	category	Category of the greeting card
	price	Price of the card in INR

Table: customer

customerid	custname
C101	Stephan
C102	Simon
C103	Eliza
C104	Maria

Table: greetingcard

cardid	customerid	category	price
GC101	C101	Festive Special	250
GC102	C102	Wedding Special	400
GC103	C104	Birthday Special	450
GC104	C102	Festive Special	500
GC105	C103	Birthday Special	300
GC106	C104	Wedding Special	600

1). Write a SQL query to display category and price of the greeting cards with price more than INR 300 or greeting cards of category 'Birthday Special'. The query result should NOT contain duplicate records.

2). Write a SQL query to display customer's id whose name contains only 5 letters or if the customers have purchased a greeting card whose price is more than INR 300. The query result may contain duplicate records.

3). Write a SQL query to display CLASSIFICATION, ID and NAME as follows:

- Display ID with the values of customer's id and greeting card's id for all rows.
- Display NAME with the values of customer's name and greeting card's category for all rows.

Solution:

1). select category,price from greetingcard where (price>300 OR category='Birthday Special');

2). select customerid from greetingcard where customerid=(select customerid from greetingcard where price>300 OR select customerid from customer where LEN(customername)=5);

3). create view ID as select customerid,cardid from greetingcard;

create view NAME as (select customername from customer AND select category from greetingcard);

Assignment-4: (Keys and ER Diagram)

We will be using the below scenario of an EasyShop Retail Store as a running example and context for most of the assignments in this course.
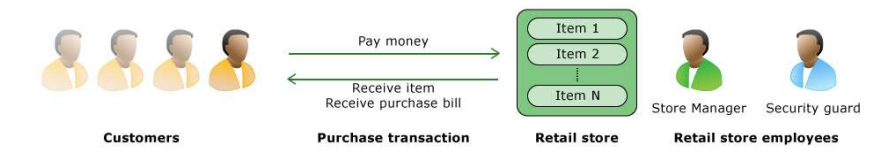
The assignments for the day must be worked out on notepad.

All relations for the retail shop application along with their attributes are mentioned below. A customer can buy multiple items and will receive different bills for each item bought. A customer can purchase at more than 1 retail outlet. Each retail store will have multiple employees. Employees can work in only 1 retail outlet. Managers are also employees. Customers living at the same address are spouses.

- **Customer** (CustId, CustType, CustName, EmailId, ContactNo, Address)
- **RetailOutlet** (ROId, Location, ManagerId)
- **Item** (ItemCode, ItemType, Desc, Price, ReorderLevel, QtyOnHand, Category)
- **Employee** (EmpId, EmpName, Designation, EmailId, ContactNo, WorksIn)
- **PurchaseBill** (BillId, RetailOutlet, Item, Customer, BillAmount, Bill Date,Quantity)

Identify **candidate key(s)**, **primary key** and **foreign key(s)** for the above relations.

Identify the possible relationships and their cardinality.

The management of EasyShop wants to identify the customer and his/her spouse who is also a customer. How do you create this relationship?

Solution:

1).

Entity	Primary Key	Candidate Key	Foreign Key
Customer	Custid	Custid,emailed	Not exists
RetailOutlet	ROid	ROid	Managerid
Item	Itemcode	Itemcode,Desc	Not exists
Employee	Empid	Empid,Emailid	Not exists
PurchaseBill	Billid	Billid	Custid,ROID,Itemcode,

2). RelationShips and Cardinality,

- WorksIn – Employee worksIn RetailOutlet, Cardinality - (Many to One)
- Buys in – Customer Buys in Retail Outlet, Cardinality – (Many to Many)
- Buys an – Customer Buys an Item, Cardinality – (Many to Many)
- Has – Customer has Purchase Bill, Cardinality –(One to Many)
- PresentIn – Item PresentIn RetailOutlet, Cardinality – (Many to Many)
- Contains – PurchaseBill Contains Item, Cardinality – (Many to Many)

3). ER-diagram,