**Tutorial 3**

**KIT712**

This week we are going to learn how to how to create queries to retrieve data from database.

**Part 1: Relational Algebra**

1. **Relational Operators**

Consider following tables:

*Order*

|  |  |  |
| --- | --- | --- |
| OrderID | CustomerID | OrderDate |
| 10308 | 2 | 1996-09-18 |
| 10309 | 37 | 1996-09-19 |
| 10310 | 77 | 1996-09-20 |

*Customer*

|  |  |  |  |
| --- | --- | --- | --- |
| CustomerID | CustomerName | ContactName | Country |
| 1 | Alfreds Futterkiste | Maria Anders | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mexico |

**Compute the following:**

1. σ OrderID=10308(Order)
2. σ OrderID>10308(Order)
3. πcustomerName(Customer)
4. πcustomerID(Customer) πcustomerID(Order)
5. πcustomerID(Customer) πcustomerID(Order)
6. πcustomerID(Customer) πcustomerID(Order)
7. πcustomerID(Order) πcustomerID(Customer)
8. OrderCustomer
9. (Order customerID Customer)))
10. **Relational Algebra Queries**

Consider the following relational model for Location database:

**lives (person\_name, street, city)**

**works(person\_name, company\_name, salary)**

**located\_in(company\_name, city)**

**manages(person\_name, manager\_name)**

The meanings of the relations are as follows:

* "lives" records the fact that person "person\_name" lives at address "street" in city "city"
* "works" records the fact that person "person\_name" works in company "company\_name" earning "salary"
* "located\_in" records the fact that company "company\_name" is located in city "city"
* "manages" records the fact that person "person\_name" is managed by manager "manager\_name"

The underline attribute(s) in each relation is the primary key. The foreign keys are:

* works.person\_name is a foreign key of relation lives,
* works.company\_name is a foreign key of relation located\_in,
* manages.manager\_name is a foreign key of relation lives,

Formulate a Relational Algebra Query on the Location database for the following:

1. Find the name of the person who live in the city “XYZ”
2. List the person name and the city they work in.
3. Find the name for all persons that **do not** work for the company "Acme"

**Part 2: SQL**

To do following tutorial questions, you need to understand your tables first. You may use following command to see the structure of the database. Open SqlDeveloper in your virtual machine and double click HR which should open a file where you can type SQL statements. Your tutor will be helping you to browse through your HR database tables.

To know the tables in your HR database, you may have to execute following query:

To display the table description, you need to use following command:

describe *tablename*

**SQL QUERIES**

**You need to write SQL queries to get the required information.**

**Single Table Queries**

1. Display the last name, job code, hire date, and employee number for each employee, with the employee number appearing first.
2. Display all unique job codes from the EMPLOYEES table.
3. Display the last name and salary of employees earning more than $12,000.
4. Display the last name and salary of employees who earn between $5000 and $12,000 and are in department 20 or 50. Label the columns Employee and Monthly Salary, respectively.
5. List the last name and department ID of all employees in departments 20, 30 or 50 in ascending alphabetical order by name.
6. Display the last name and department number of all employees in departments 20 or 50 in ascending alphabetical order by name
7. List the last name and hire date for all employees who are hired in 2005.
8. Create a query that displays the last name and salary of employees who earn more than an amount the user specifies after a prompt.
9. Create a query that prompt the user for a manager ID and generates the employee ID, last name, salary and department for that manager’s employees.
10. Display minimum and maximum salary given in the company.
11. For each employee, list the last name and calculate the number months between today and the date on which the employee was hired. Label the column months\_worked. Round the number of months up to the closest whole number.

**Solution:**

**-- Create Order table**

**CREATE TABLE OrderTable (**

**OrderID INT PRIMARY KEY,**

**CustomerID INT,**

**OrderDate DATE**

**);**

**-- Create Customer table**

**CREATE TABLE Customer (**

**CustomerID INT PRIMARY KEY,**

**CustomerName VARCHAR(100),**

**ContactName VARCHAR(100),**

**Country VARCHAR(50)**

**);**

**-- Create Location database tables**

**CREATE TABLE lives (**

**person\_name VARCHAR(100),**

**street VARCHAR(100),**

**city VARCHAR(50),**

**PRIMARY KEY (person\_name)**

**);**

**CREATE TABLE works (**

**person\_name VARCHAR(100),**

**company\_name VARCHAR(100),**

**salary INT,**

**PRIMARY KEY (person\_name),**

**FOREIGN KEY (person\_name) REFERENCES lives(person\_name)**

**);**

**CREATE TABLE located\_in (**

**company\_name VARCHAR(100),**

**city VARCHAR(50),**

**PRIMARY KEY (company\_name)**

**);**

**CREATE TABLE manages (**

**person\_name VARCHAR(100),**

**manager\_name VARCHAR(100),**

**PRIMARY KEY (person\_name),**

**FOREIGN KEY (manager\_name) REFERENCES lives(person\_name)**

**);**

**-- Create Employees table**

**CREATE TABLE employees (**

**employee\_id INT PRIMARY KEY,**

**last\_name VARCHAR(100),**

**job\_id VARCHAR(50),**

**hire\_date DATE,**

**salary DECIMAL(10,2),**

**department\_id INT,**

**manager\_id INT**

**);**

**-- Insert data into Order table**

**INSERT INTO OrderTable VALUES (10308, 2, TO\_DATE('1996-09-18', 'YYYY-MM-DD'));**

**INSERT INTO OrderTable VALUES (10309, 37, TO\_DATE('1996-09-19', 'YYYY-MM-DD'));**

**INSERT INTO OrderTable VALUES (10310, 77, TO\_DATE('1996-09-20', 'YYYY-MM-DD'));**

**-- Insert data into Customer table**

**INSERT INTO Customer VALUES (1, 'Alfreds Futterkiste', 'Maria Anders', 'Germany');**

**INSERT INTO Customer VALUES (2, 'Ana Trujillo Emparedados y helados', 'Ana Trujillo', 'Mexico');**

**INSERT INTO Customer VALUES (3, 'Antonio Moreno Taquería', 'Antonio Moreno', 'Mexico');**

**-- Insert data into lives table**

**INSERT INTO lives VALUES ('John Doe', '123 Elm St', 'XYZ');**

**INSERT INTO lives VALUES ('Jane Smith', '456 Oak St', 'ABC');**

**INSERT INTO lives VALUES ('Alice Johnson', '789 Pine St', 'XYZ');**

**-- Insert data into works table**

**INSERT INTO works VALUES ('John Doe', 'Acme', 50000);**

**INSERT INTO works VALUES ('Jane Smith', 'BetaCorp', 60000);**

**INSERT INTO works VALUES ('Alice Johnson', 'Gamma Inc', 70000);**

**-- Insert data into located\_in table**

**INSERT INTO located\_in VALUES ('Acme', 'XYZ');**

**INSERT INTO located\_in VALUES ('BetaCorp', 'ABC');**

**INSERT INTO located\_in VALUES ('Gamma Inc', 'XYZ');**

**-- Insert data into manages table**

**INSERT INTO manages VALUES ('John Doe', 'Alice Johnson');**

**INSERT INTO manages VALUES ('Jane Smith', 'John Doe');**

**INSERT INTO manages VALUES ('Alice Johnson', 'Jane Smith');**

**-- Insert data into employees table**

**INSERT INTO employees VALUES (101, 'Smith', 'IT\_PROG', TO\_DATE('2005-06-01', 'YYYY-MM-DD'), 12000, 20, 201);**

**INSERT INTO employees VALUES (102, 'Johnson', 'HR\_REP', TO\_DATE('2008-09-15', 'YYYY-MM-DD'), 8000, 50, 202);**

**INSERT INTO employees VALUES (103, 'Williams', 'SA\_MAN', TO\_DATE('2010-12-10', 'YYYY-MM-DD'), 15000, 30, 203);**

**INSERT INTO employees VALUES (104, 'Brown', 'IT\_PROG', TO\_DATE('2003-03-25', 'YYYY-MM-DD'), 10000, 20, 201);**

**INSERT INTO employees VALUES (105, 'Davis', 'SA\_REP', TO\_DATE('2005-07-20', 'YYYY-MM-DD'), 7000, 50, 204);**

**-- SQL Queries**

**-- 1. Display the last name, job code, hire date, and employee number for each employee**

**SELECT employee\_id, last\_name, job\_id, hire\_date FROM employees;**

**-- 2. Display all unique job codes from the EMPLOYEES table**

**SELECT DISTINCT job\_id FROM employees;**

**-- 3. Display the last name and salary of employees earning more than $12,000**

**SELECT last\_name, salary FROM employees WHERE salary > 12000;**

**-- 4. Display the last name and salary of employees who earn between $5000 and $12,000 and are in department 20 or 50**

**SELECT last\_name AS Employee, salary AS "Monthly Salary" FROM employees WHERE salary BETWEEN 5000 AND 12000 AND department\_id IN (20, 50);**

**-- 5. List the last name and department ID of all employees in departments 20, 30 or 50 in ascending alphabetical order by name**

**SELECT last\_name, department\_id FROM employees WHERE department\_id IN (20, 30, 50) ORDER BY last\_name ASC;**

**-- 6. Display the last name and department number of all employees in departments 20 or 50 in ascending alphabetical order by name**

**SELECT last\_name, department\_id FROM employees WHERE department\_id IN (20, 50) ORDER BY last\_name ASC;**

**-- 7. List the last name and hire date for all employees who are hired in 2005**

**SELECT last\_name, hire\_date FROM employees WHERE EXTRACT(YEAR FROM hire\_date) = 2005;**

**-- 8. Create a query that displays the last name and salary of employees who earn more than an amount the user specifies after a prompt**

**SELECT last\_name, salary FROM employees WHERE salary > :user\_input;**

**-- 9. Create a query that prompts the user for a manager ID and generates the employee ID, last name, salary, and department for that manager’s employees**

**SELECT employee\_id, last\_name, salary, department\_id FROM employees WHERE manager\_id = :manager\_id;**

**-- 10. Display minimum and maximum salary given in the company**

**SELECT MIN(salary) AS min\_salary, MAX(salary) AS max\_salary FROM employees;**

**-- 11. For each employee, list the last name and calculate the number of months between today and the date on which the employee was hired**

**SELECT last\_name, CEIL(MONTHS\_BETWEEN(SYSDATE, hire\_date)) AS months\_worked FROM employees;**