**1.Introduction to Relational Databases: Explain why using a relational database is beneficial for storing information about products, customers, and sales in the context of a retail business. Provide at least two advantages.**

Using a relational database in a retail business is beneficial because:

* **Data Organization and Structure:** Data is stored in well-defined tables, making it easier to manage large volumes of information such as product inventory, customer profiles, and sales transactions.
* **Easy Data Retrieval and Analysis:**With SQL queries, businesses can quickly retrieve insights like top-selling products, customer purchase history, or sales trends.

**2. Data Integration and ETL: The company has decided to integrate data from multiple sources, including online sales, in-store transactions, and customer feedback. Outline the steps you would take in the ETL process (Extract, Transform, Load) to ensure the data is consolidated and ready for analysis. Highlight any potential challenges you might encounter.**

Data Integration and ETL:

To combine data from online sales, in-store transactions, and customer feedback, the ETL process is used to prepare data for analysis.

Steps in ETL:

1. Extract: Collect data from all sources (e.g., e-commerce, POS, feedback forms).
2. Transform: Clean and standardize data, remove duplicates, fix errors, and integrate it into a common format.
3. Load: Store the transformed data into a central database or data warehouse for analysis.

Potential Challenges:

* Inconsistent or missing data
* Different data formats
* Large data volumes
* Complex integration between sources

**3.SQL Fundamentals: Write an SQL query to retrieve the names and prices of all products in the "Electronics" category that have a price greater than $500. Include the product name and price columns in your result set.**

SELECT product\_name, price

FROM products

WHERE category = 'Electronics'

AND price > 500;

**4.Advanced SQL: The marketing team is interested in identifying customers who have made both online and in-store purchases. Write an SQL query that retrieves the customer IDs of individuals who have transactions in both the "OnlineTransactions" and "InStoreTransactions" tables.**

SELECT DISTINCT o.customer\_id

FROM OnlineTransactions AS o

JOIN InStoreTransactions AS i

ON o.customer\_id = i.customer\_id;

**5.Write SQL queries to create a database having relationships between tables. Insert data given below also.**

**Create Database**

CREATE DATABASE CompanyDB;

USE CompanyDB;

**Create Tables**

CREATE TABLE Employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

phone\_number VARCHAR(20),

hire\_date DATE,

job\_id INT,

salary DECIMAL(10,2),

department\_id INT,

FOREIGN KEY (job\_id) REFERENCES Jobs(job\_id),

FOREIGN KEY (department\_id) REFERENCES Departments(department\_id)

);

**Departments**

CREATE TABLE Departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(50),

manager\_id INT,

location\_id INT,

FOREIGN KEY (manager\_id) REFERENCES Employees(employee\_id),

FOREIGN KEY (location\_id) REFERENCES Locations(location\_id)

);

#### ****Jobs****

CREATE TABLE Jobs (

job\_id INT PRIMARY KEY,

job\_title VARCHAR(50),

min\_salary DECIMAL(10,2),

max\_salary DECIMAL(10,2)

);

**Locations**

CREATE TABLE Locations (

location\_id INT PRIMARY KEY,

street\_address VARCHAR(100),

postal\_code VARCHAR(10),

city VARCHAR(50),

state\_province VARCHAR(10),

country\_id INT,

FOREIGN KEY (country\_id) REFERENCES Countries(country\_id)

);

**Countries**

CREATE TABLE Countries (

country\_id INT PRIMARY KEY,

country\_name VARCHAR(50)

);

**Job\_History**

CREATE TABLE Job\_History (

employee\_id INT,

start\_date DATE,

end\_date DATE,

job\_id INT,

department\_id INT,

PRIMARY KEY (employee\_id, start\_date),

FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id),

FOREIGN KEY (job\_id) REFERENCES Jobs(job\_id),

FOREIGN KEY (department\_id) REFERENCES Departments(department\_id)

);

**Insert Data**

#### ****Countries****

INSERT INTO Countries VALUES

(1, 'Australia'),

(2, 'Vietnam'),

(3, 'Spain');

**Locations**

INSERT INTO Locations VALUES

(10, '123 Collins St', '3000', 'Melbourne', 'VIC', 1),

(20, '222 Bourke St', '3000', 'Melbourne', 'VIC', 1),

(30, '555 Swanston St', '3000', 'Melbourne', 'VIC', 1);

**Jobs**

INSERT INTO Jobs VALUES

(10, 'Den Manager', 120000, 150000),

(22, 'Accountant', 60000, 80000),

(33, 'Programmer', 60000, 80000),

(45, 'Senior Programmer', 70000, 120000);

**Employees**

INSERT INTO Employees VALUES

(50, 'Adam', 'Smith', '1234', '2009-10-26', 22, 66000, 2),

(66, 'Tom', 'Moosa', '1235', '2016-10-12', 10, 140000, 1),

(10, 'Jonny', 'Deans', '1236', '2015-09-09', 33, 70000, 1),

(12, 'Adam', 'Jones', '1247', '2019-08-08', 10, 138000, 1),

(18, 'Joseph', 'Ryan', '1277', '2020-05-05', 10, 150000, 3);

**Departments**

INSERT INTO Departments VALUES

(1, 'IT Services', 12, 10),

(2, 'Accounting', 66, 20),

(3, 'Human Resource', 18, 30);

**Job\_History**

INSERT INTO Job\_History VALUES

(10, '2001-01-01', '2002-10-04', 33, 1),

(10, '2002-11-04', '2002-08-20', 33, 1),

(12, '1998-01-01', '2003-05-10', 33, 1),

(12, '2003-06-10', '2004-06-10', 33, 1),

(12, '2004-07-10', '2009-07-08', 33, 1);