1. ***If you were designing a Web-based system to make airline reservations and to sell airline tickets, which DBMS Architecture would you choose from Centralized and Client/Server Architectures? Why? And why would the other architectures not be a good choice? (8 pts.)***

   For a web based air reservations and sales system, I would recommend to use a Three-Tier Architecture over a Two-Tier Architecture. Due to the following reasons:

   - A two-tier Architecture comprises of set of Servers and clients on a single network. This could impact to the security of the database, as Database Server is directly interacting with the Client. This can be solved by using a three-tier architecture, which includes a GUI, Web Application and the DBMS. Here clients have access only to the GUI which sends data to Web Application to process the same. Only Web Application has access to the DBMS System.
   - Considerable high performance can be achieved in a three-tier as network buses are not always busy. Along with that GUI helps in caching data, thus decreasing the load on web application server.
   - A three-tier architecture is easy to scale. Multiple web application server can be added to handle extreme loads on the primary web application server. Each tier in this architecture can be horizontally.

   I would not go for a two-tier architecture due to following reasons:

   - With increase in clients, the performance of the system takes a hit. That is both network is overloaded as well as requests to the Servers.
   - Security is a concern as the clients are in direct contact with the servers.

2. ***Discuss the main categories of data models. What are the basic differences between the relational model, the object model, and the XML model? (8 pts.)***

   There are three main categories of data models. They are:

   - High level or Conceptual Data Models

     It is a map of concepts and their relationships used to build database. Concepts and relationships that simple user could understand. They mainly comprises of entities and the relationship between them.

   - Representational or Implementation Data Models

     It is the intermediate data model, where definition of the entities like their attributes, constrains and mapping along with their data types are defined. It helps an end user to understand the database structure in a graphical method.

   - Low level or Physical Data Models

     It is a data model that describes the actual implementation of the database using a specific database management system. This model is particularly implemented by DBAs and developers.

The basic differences between the relational model, the object model and XML model are:

Relational Model	Object Model	XML Model
representation in which the database is represented in form of collections of tables.	Representation in which the database is represented in form of objects, properties and their operations.	Representation of data in an hierarchical tree formats.
Each table is stored in a separate file.	Objects with similar structure are pushed in a class structure.	Combination of database concepts and documentation representational models.

3. ***Identify the relation name, tuple, attributes, degree from below database table. (4 pts.)***

EMPLOYEE

Employee_Name	Employee_Id	Department	Phone_No
John Karter	12789876	Sales	5672345677
Sarah Hart	14569878	Marketing	4089733403
Edward Smith	13425235	IT	6823331233

The relation name is EMPLOYEE
The tuples are:

John Karter	12789876	Sales	5672345677
Sarah Hart	14569878	Marketing	4089733403
Edward Smith	13425235	IT	6823331233

The attributes are: Employee_Name, Employee_Id, Department, Phone_No
The degree of the above database table is 4 as there are four attributes that define the table.

4. ***Define foreign key. What is this concept used for? (4 pts.)***

It is a reference key of one table that refers to a primary key of another table. Basically, they are pointers in a table that links records of another table.

The concepts used for foreign key are:

- It is used to link two tables. For instance, Entity Student would always have a link to Entity Department in an University Database.
- Foreign Keys provide relation between tables and this makes the database powerful.
- With the help of foreign keys, we could prevent actions that could destroy the links between these tables.
- With the help of foreign keys, prevents invalid data from being inserted into the foreign key column as it needs to be present in the mapped table.

5. ***Consider each of the following Update operations is applied directly to the below database (see next page). Discuss all integrity constraints violated by each operation, if any, and the different ways of enforcing these constraints: (30 pts.)***

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

a. ***Insert < 'Robert', 'F', 'Scott', '943775543', '21-JUN-42', '2365 Newcastle Rd, Bellaire, TX', M, 58000, '888665555', 1 > into EMPLOYEE.***

   No violation

b. ***Insert < 'ProductA', 4, 'Bellaire', 2 > into PROJECT.***

   This violates the foreign referential constrain. There is no Department with Dnum = 2

c. ***Insert < 'Production', 4, '943775543', '01-OCT-88' > into DEPARTMENT.***

   This violates primary key constrain. There is already a Department with Dnumber = 4. It will also violate the foreign referential constrain, as there is no SSN of 943775543 in the EMPLOYEE table.

d. ***Insert < '677678989', null, '40.0' > into WORKS_ON.***

   This violates foreign referential constrain. Here, there is no SSN of 677678989 in the EMPLOYEE table which Essn is indexed to. Also, null for Pno cannot be null as both Essn and Pno form the candidate key for the table.

e. ***Insert < '453453453', 'John', M, '12-DEC-60', 'SPOUSE' > into DEPENDENT.***
No violation

f. ***Delete the WORKS_ON tuples with ESSN= '333445555'.***
No violation

g. ***Delete the EMPLOYEE tuple with SSN= '987654321'.***
This violates foreign referencial constrain. The reference of SSN = 987654321 has been mapped to couple of records in WORKS_ON Entity with attribute Essn and a record entry in DEPENDENT Entity with attribute Essn.

h. ***Delete the PROJECT tuple with PNAME= 'ProductX'.***
This violates foreign referential constrain. The Pnumber of Project with PNAME = 'ProductX' has been used in entity WORKS_ON.Pno attribute

i. ***Modify the SUPERSSN attribute of the EMPLOYEE tuple with SSN= '999887777' to '943775543'.***
This violates foreign referential constrain as their exists no EMPLOYEE with SSN = 943775543

j. ***Modify the HOURS attribute of the WORKS_ON tuple with ESSN= '999887777' and PNO= 10 to '5.0'.***
No violation.

6. ***Consider the following relations for a database that keeps track of business trips of salespersons in a sales office: (8 pts.)***

SALESPERSON (SSN, Name, Start_Year, Dept_No)
TRIP (SSN, From_City, To_City, Departure_Date, Return_Date, Trip_ID)
EXPENSE (Trip_ID, Account#, Amount)

Specify 4 foreign keys for this schema, stating any assumptions you make.

1. 'SSN' attribute of 'TRIP' Entity is a foreign key mapped to 'SSN' attribute of 'SALESPERSON' Entity.
2. 'TRIP_ID' attribute of 'EXPENSE' Entity is a foreign key mapped to 'TRIP_ID' attribute of 'TRIP' Entity.
3. When DEPARTMENT(Dept_No, Dept_Name, Location) is an Entity, then 'Dept_No' attribute of 'SALESPERSON' Entity is a foreign key mapped to 'Dept_No' attribute of 'DEPARTMENT' Entity.
4. When ACCOUNTS(AccountNo, Account_Type, Bank_Code, Bank_Account_No) is an Entity, then 'Account#' attribute of 'EXPENSE' Entity is a foreign key mapped to 'AccountNo' of 'ACCOUNTS' Entity.

7. ***Consider the relation CLASS (Course#, Univ_Section#, InstructorName, Semester, BuildingCode, Room#, TimePeriod, Weekdays, CreditHours). This represents classes taught in a university with unique Univ_Section#. Give what you think should be at least 4 candidate keys and write in your own words under what constraints each candidate key***

would be valid.(Note: The values of the Semester attribute include the year; for example "Spring/94" or "Fall/93") (12 pts.)

a. Univ_Section#
   This is the primary key of the table CLASS, thus unique for all its records. Every primary key is also a candidate key.

b. Course#, Univ_Section#
   Here Univ_Section# is primary key of the table. This makes all the records of the table unique. Any attribute can be added to it that could be a generate a proper candidate key. This Course# along with Univ_Section# makes it a good candidate key.

c. Room#, Univ_Section#
   Here Univ_Section# is primary key of the table. This makes all the records of the table unique. Any attribute can be added to it that could be a generate a proper candidate key. This Room# along with Univ_Section# makes it a good candidate key.

d. Semester, Univ_Section#
   Here Univ_Section# is primary key of the table. This makes all the records of the table unique. Any attribute can be added to it that could be a generate a proper candidate key. This Semester along with Univ_Section# makes it a good candidate key.

8. ***Design an Automobile showroom database. Specify the possible attributes for each of the below tables. You can add more tables if needed. Also specify the primary keys and the foreign keys for the table. You don't have to enter any data. Please mention any assumptions you make. (10 pts.)***

   Note: Mention at least 5 attributes for each of the tables.
   Database Name: Automobile Table
   Names: Employee, Customer, Car, Buy

EMPLOYEE

EMP_ID	FNAME	LNAME	DOB	JOIN_DATE	POSITION	PHONE

CUSTOMER

CUST_ID	CUST_NAME	PHONE	EMAIL	ADDRESS	DOB	GENDER

CAR

CAR_ID	CAR_NAME	MANUFACTURER	MODEL	COLOR

BUY

BUY_ID	CUST_ID	CAR_ID	AMOUNT	PURSCHASE_DATE	SOLD_BY

9. ***Specify all the referential integrity constraints for the AIRLINE relational database schema shown below which describes a database for airline flight information. You may***

write referential integrity constraint as R.A --> S (or R.(X) --> T) whenever attribute A (or the set of attributes X) of relation R form a foreign key that references the primary key of relation S (or T). (16 pts.)

AIRPORT

Airport_code	Name	City	State

FLIGHT

Flight_number	Airline	Weekdays

FLIGHT_LEG

Flight_number	Leg_number	Departure_airport_code	Scheduled_departure_time
		Arrival_airport_code	Scheduled_arrival_time

FARE

Flight_number	Fare_code	Amount	Restrictions

AIRPLANE_TYPE

Airplane_type_name	Max_seats	Company

CAN_LAND

Airplane_type_name	Airport_code

AIRPLANE

Airplane_id	Total_number_of_seats	Airplane_type

- FLIGHT_LEG.Flight_number -> FLIGHT
- FLIGHT_LEG.Departure_airport_code -> AIRPORT
- FLIGHT_LEG.Arrival_airport_code -> AIRPORT
- FARE.Flight_number -> FLIGHT
- CAN_LAND.Airplane_type_name -> AIRPLANE_TYPE
- CAN_LAND.Airport_code -> AIRPORT
- AIRPLANE.Airplane_Type -> AIRPLANE_TYPE

References

- https://en.wikipedia.org/wiki/Conceptual_schema
- https://www.guru99.com/data-modelling-conceptual-logical.html
- https://www.softwaretestingclass.com/what-is-difference-between-two-tier-and-three-tier-architecture/
- https://www.w3schools.com/sql/sql_foreignkey.asp