The University of Melbourne
School of Computing and Information Systems
INFO20003 Database Systems

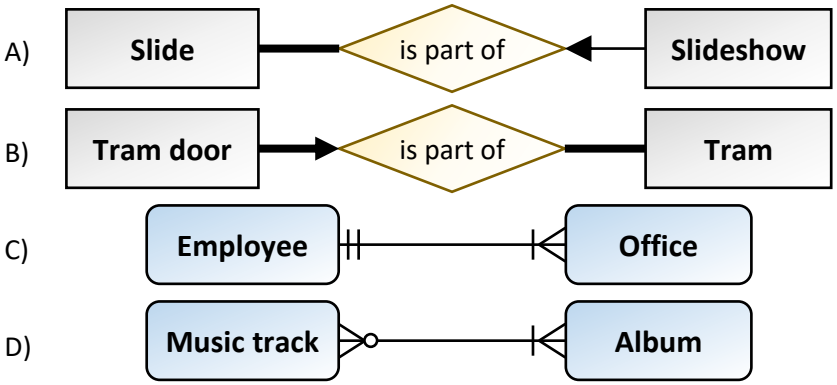
Miscellaneous practice questions

Data modelling

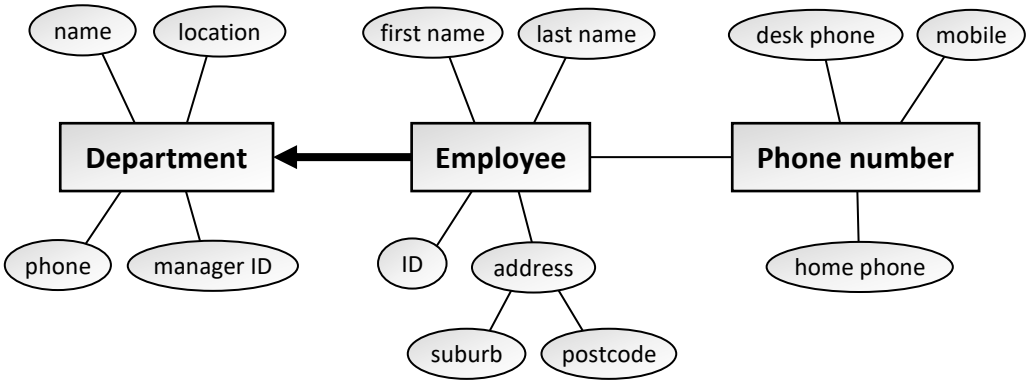
1. Select appropriate MySQL data types for the following pieces of information:

   a. A person's last name
   b. The number of items purchased in a particular transaction
   c. The average number of items purchased
   d. A phone number
   e. When a tap-and-go purchase took place
   f. Whether an order has been completed or not
   g. The unit price of an item at a convenience store
   h. The unit price of a vehicle at a car dealership
   i. A person's gender, stored according to Australian Government guidelines – M (Male), F (Female) or X (Indeterminate/Intersex/Unspecified)
   j. A person's Aboriginal and Torres Strait Islander status – Aboriginal, Torres Strait Islander, both, or neither

2. Assuming that common sense applies, which **one** of the following ER model fragments is most believable?



3. A student was asked to model a typical employee-department scenario by drawing a conceptual ER model in Chen's notation. Their model, shown below, has a number of mistakes. Identify the mistakes and redraw the model correctly.



4. Imagine you are developing a database to store information about marriages.

   a. Using Chen's notation, draw a "person" entity with appropriate attributes and a "married to" relationship that records who is currently married to whom. What type of relationship is this?
   b. Based on your answer to part a, draw a model that stores not only current marriages, but also all previous marriages.
   c. Based on your answer to part b, draw a model that stores all current and previous marriages, including the date of marriage and, for previous marriages, date of dissolution.
   d. Can your answer to part c handle the situation where the same two people marry, divorce and later marry again? What do you need to change to handle this?
   e. In certain societies (both historical and modern), a person can be married to many people at the same time. Draw another model to handle this scenario, based on your answer to part d.
   f. Resolve all five Chen's notation models into physical models using Crow's foot notation. Take particular care that your physical models for parts d and e can handle the scenarios explained in those parts.

Relational algebra

5. The following tables are part of a database for a role-playing game:

   player (playerid, playername, experience)
   playeritem (playerid, itemid, quantity) — playerid: FK, itemid: FK
   item (itemid, itemname, value, weight, colour)

   Write relational algebra expressions to answer the following questions:

   a. List all the possible experience levels of players.
   b. Show all information about gold-coloured items with a value of 20 coins.
   c. List the names of players who have bought a "Vine Knife".
   d. List the names and values of items bought by players with experience level "Adventurer" who are not named "coulter".
   e. Find names which are shared by at least one player and at least one item.

SQL

6. Consider the following tables:

   Employee (EmployeeID, Name, Phone, DepartmentID) — DepartmentID: FK
   Department (DepartmentID, DepartmentName, DepartmentFloor)

   This SQL query returns all employees who work on the fifth floor:

```
SELECT Name
FROM Employee NATURAL JOIN Department
WHERE DepartmentFloor = 5;
```

   Rewrite the query using the following SQL techniques:

   a. `INNER JOIN`
   b. Cross product
   c. Outer join
   d. `IN`
   e. `EXISTS`
   f. `ANY`

7. Which **two** of these fragments **cannot** appear as part of a SELECT query?

   a. `SELECT COUNT(*)`
   b. `WHERE COUNT(*) > 1`
   c. `GROUP BY COUNT(*)`
   d. `HAVING COUNT(*) > 1`
   e. `ORDER BY COUNT(*)`

8. Consider the following table definitions:

   Employee (EmployeeID, EmployeeName, Phone, BuildingID (FK), RoomNumber)
   Building (BuildingID, BuildingName, NumFloors)

   Not every employee is located in a room; the BuildingID and RoomNumber columns on Employee may be NULL.

   Explain what is wrong with the following SQL queries, and suggest a correction. (Some of the queries contain invalid syntax, and others are technically valid but conceptually incorrect.)

   a.
   ```
   SELECT *
   FROM Building
     OUTER JOIN Employee ON Building.BuildingID = Employee.BuildingID;
   ```

   b.
   ```
   SELECT EmployeeName, BuildingName
   FROM Building INNER JOIN Employee;
   ```

   c.
   ```
   SELECT EmployeeName, BuildingID
   FROM Building
     INNER JOIN Employee ON Building.BuildingID = Employee.BuildingID;
   ```

   d.
   ```
   SELECT EmployeeName, BuildingID, COUNT(EmployeeID)
   FROM Building NATURAL JOIN Employee
   GROUP BY BuildingID;
   ```

   e. *(We were asked to find how many employees are in each occupied single-storey building.)*
   ```
   SELECT BuildingName, COUNT(EmployeeID)
   FROM Building NATURAL JOIN Employee
   GROUP BY BuildingID
   HAVING NumFloors = 1;
   ```

   f. *(We were asked to find the number of employees in every occupied building, placing taller buildings before shorter buildings in the result set.)*
   ```
   SELECT BuildingName, COUNT(*)
   FROM Building NATURAL JOIN Employee
   ORDER BY NumFloors DESC
   GROUP BY BuildingName;
   ```

   g. *(We were asked to find all buildings which are not occupied by any employees.)*
   ```
   SELECT BuildingName
   FROM Building
   WHERE BuildingID NOT IN (SELECT BuildingID
                            FROM Employee);
   ```

Indexes and query cost

9. Consider the following table:

   Employee (EmployeeID, Name, Age, DepartmentID)

   Suppose the following three queries are executed frequently on this table:

   i.
   ```
   SELECT Name
   FROM Employee
   WHERE Age > 30 AND DepartmentID = 5;
   ```

   ii.
   ```
   SELECT EmployeeID, Name
   FROM Employee
   WHERE Age = 65;
   ```

   iii.
   ```
   SELECT Name, Age
   FROM Employee
   WHERE DepartmentID = 7;
   ```

   Out of the above queries, which ones (if any) could potentially make use of:

   a. A clustered B-tree index on Age
   b. A hash index on Age
   c. A hash index on DepartmentID
   d. A clustered B-tree index on (DepartmentID, EmployeeID)
   e. A hash index on (EmployeeID, DepartmentID)
   f. An unclustered B-tree index on (DepartmentID, Age)

10. The Employee table from question 9 has 50 data pages, with 20 tuples per page. Employees are between 25 and 65 years old, and there are 20 different departments. All indexes have 10 index pages.

    a. Calculate the reduction factors for each of the three queries.
    b. Compute the estimated cost of the best plan for each query, assuming that an unclustered B-tree index on (DepartmentID, Age) is the only index available. Discuss and calculate alternative plans.
    c. Compute the estimated cost of the best plan for each query, assuming that a clustered B-tree index on (Age, DepartmentID) is the only index available. Discuss and calculate alternative plans.

Query optimisation

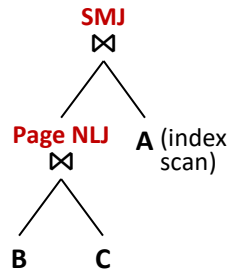
11. Consider the relations:

    A (Aid, …) – 2500 tuples, 10 tuples per page
    B (Bid, Aid (FK), …) – 300 tuples, 50 tuples per page
    C (Cid, Bid (FK), …) – 2000 tuples, 20 tuples per page

    A clustered B+ tree index exists on the Aid column in relation A, with 100 index pages.

    Assume that two passes are required to sort. For all join results, 10 tuples can be stored per page.

    Calculate the cost of the following plan, and determine the final result size.

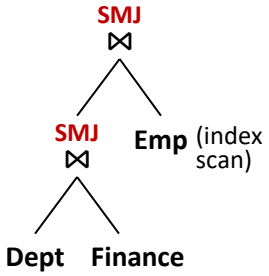


12. Consider the following SQL statement:

```
SELECT *
FROM Dept, Emp, Finance
WHERE Dept.did = Finance.did AND Dept.did = Emp.did;
```

The sorting of a relation can be done in 2 passes. A page holds 10 tuples. There are 1000 employees, with a total of 100 departments. Each department has a corresponding financial record. There is a clustered B+ tree index on `Emp.did` which contains 50 pages.

Calculate the cost of the following plan, and determine the final result size.



Normalisation

13. Consider the following relation:

MealsOrdered (OrderID, CustomerID, CustomerName, DishID, DishName, DishPrice)

It has the following functional dependencies:

OrderID → CustomerID, CustomerName
CustomerID → CustomerName
DishID → DishName, DishPrice

a. Label these functional dependencies as partial, transitive or neither.
b. Normalise this relation to second normal form. Write down the functional dependencies that remain.
c. Normalise this relation to third normal form.

14. Selected rows from the OrderItem table of an online retailer are shown below.

OrderID	ItemID	CustomerID	CustomerPostcode	ItemQuantity	CanDispatchFrom
4018	161	191	3053	6	Truganina, Hallam
4022	228	196	3212	1	Somerton
4033	525	25	3124	2	Somerton, Hallam

(OrderID, ItemID) is the primary key. Several functional dependencies exist on this table:

CustomerID → CustomerPostcode
OrderID → CustomerID
OrderID, ItemID → ItemQuantity, CanDispatchFrom

Normalise the table to third normal form. Write the normalised tables in a textual format, as in:

TableName (PrimaryKey, Column, ForeignKey^{FK})
AnotherTable (PrimaryKey, Column, AnotherColumn)

15. The following table stores information about players in a sports league. Five of the rows are shown below – the actual table has many more rows.

Player ID	Player Name	Player Shirt Size	Team Name	Team Mascots	Team Home Ground	Home Ground Location
23	Sam Binns	XL	Carlton Colts	Dragon	East Park	Kew
24	Taylor Colosimo	M	Port Melbourne Pintos	Cat, Bear	South Arena	Elwood
26	Hamish Baker	M	Carlton Colts	Dragon	East Park	Kew
27	Xiaowen Zhang	L	Richmond Racehorses	Emu, Koala	East Park	Kew
30	Luca Garzon	L	Port Melbourne Pintos	Cat, Bear	South Arena	Elwood

a. Identify the violations of first normal form (1NF), second normal form (2NF), and third normal form (3NF). For each violation, clearly state which normal form is violated.
b. Normalise the table into 3NF. Write the normalised tables in a textual format, as in:

TableName (PrimaryKey, Column, ForeignKey^{FK})
AnotherTable (PrimaryKey, Column, AnotherColumn)

Database administration

16. Consider the *original, non-normalised* table from question 15. It was created using the following SQL statement:

```
CREATE TABLE Player (
  PlayerID INT NOT NULL,
  PlayerName VARCHAR(50) NOT NULL,
  PlayerShirtSize VARCHAR(4),
  TeamName VARCHAR(40) NOT NULL,
  TeamMascots VARCHAR(40),
  TeamHomeGround VARCHAR(40),
  HomeGroundLocation VARCHAR(16),
  PRIMARY KEY (PlayerID)
);
```

Currently, there are 500 players in the sports league. It is predicted that 200 players will join the league, and 80 will leave, every year for the next 5 years.

Assuming that an INT value takes up 4 bytes, and that on average, a VARCHAR(*n*) value takes up $\frac{n}{2}$ bytes, calculate the size of the player table (a) now, and (b) after 5 years.

17. Sprint Rail runs a train network that operates from 5am to midnight every day. It relies on antiquated and unreliable computer equipment, and in the all-too-frequent event of a failure, it needs to be able to restore data as quickly as possible. Data changes from hour to hour and the operators need to be able to restore the system to a state that is no more than 1 hour old.

Last year, an electrical fault at the Sprint Rail control centre caused the server where the backups are stored to short out. The server was destroyed, together with all the files stored on it. Luckily the main systems were running smoothly at the time, but the managers were concerned that they did not have a plan B in case a catastrophe occurred.

Suggest a backup plan for Sprint Rail. Give times when backups should be taken, and decide whether the backups should be logical or physical; online or offline; onsite or offsite; and full or incremental. Your plan could include different backup configurations at different times.

Distributed databases

18. A distributed database system is most commonly implemented using either replication or horizontal partitioning.

    a. Wikipedia, the user-editable encyclopedia, uses a MySQL database as its main data store. The master database is stored in the United States, and synchronised replication is performed to a handful of other data centres. Discuss why you think Wikipedia uses a distributed database as opposed to a centralised database. Also discuss why a replication approach might have been selected for Wikipedia.
    b. As of 2018, Facebook continues to use MySQL as an essential part of its database infrastructure. If you had to design a distributed relational database for a worldwide social networking site like Facebook, would you select replication or horizontal partitioning? Discuss.

Data warehousing

19. Suppose you have been hired by a furniture wholesaler to develop a data warehouse. (A furniture wholesaler buys or imports furniture in bulk from manufacturers and sells it to showrooms and retailers.)

    The wholesaler is looking to expand its business within Australia. Some of the key performance indicators the CEO is interested in are monthly revenue growth, number of new customers by month, average number of products sold by each state sales office, and total leads generated at each office. Managers also keep a close eye on which product types (for example, sofas, desks, or outdoor seating) are performing well or badly.

    a. Identify the facts and dimensions that might be relevant to this business.
    b. For each dimension, suggest an appropriate grain.
    c. Draw a simple star schema for this data warehouse using crow's foot notation. You do not need to indicate NULL/NOT NULL nor show data types. You might need to add some attributes to the dimension tables (for example, customer name).

Transactions and NoSQL

20. Group the following statements under the headings "transactional databases", "NoSQL databases", "both" or "neither".

    a. The state of the system could change over time without input.
    b. Work being undertaken by one user doesn't interfere with the work of other users.
    c. The ACID principles apply.
    d. The system is designed for high-level strategic decision making, rather than everyday operational needs.
    e. Many systems of this type are designed with Big Data storage in mind.
    f. The user may have to wait for other users to complete their work before certain data can be retrieved.
    g. The Schema on Write approach is typically used.

h. The CAP Theorem applies.
i. It is generally less complex to create a distributed system of this type.
j. Physical backups can be carried out while the server is not running.
k. The BASE principles apply.

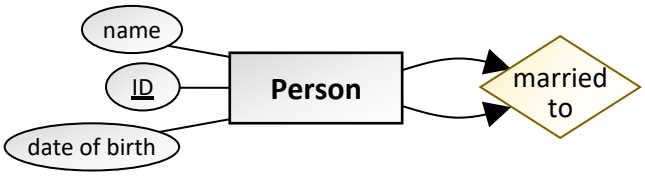
Answers

Full solutions are not provided. You're encouraged to form a small study group and work together to solve these problems and discuss your answers.

1. Question for discussion
2. B
3.
   - The lozenges that indicate relationships have been omitted.
   - A foreign key (manager ID) has wrongly been included in this conceptual model.
   - Primary keys have not been underlined.
   - "Phone number" has been incorrectly modelled as a separate entity, instead of attributes on the Employee entity.
   - The "address" composite attribute only contains two pieces of information – suburb and postcode. The actual address is not stored.

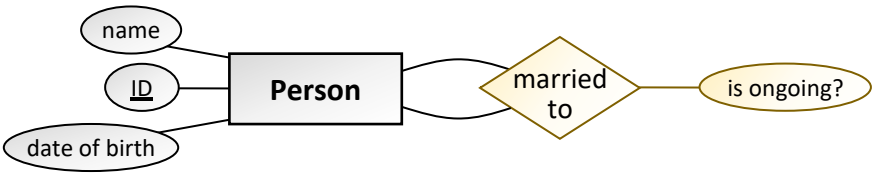
   Redrawn model is a question for discussion
4. Note that the following answers are only one possible way of solving these scenarios.

   a.

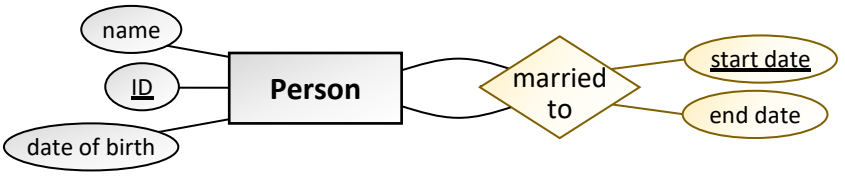


   Unary relationship

   b.

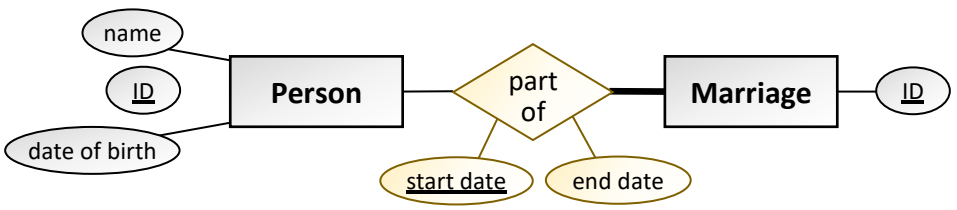


   c.



   d. "start date" needs to be a key attribute

   e.



   Making "start date" a key is necessary if a person may leave and later re-enter a marriage.

   f. Question for discussion

5. a. $\pi_{\text{experience}}(\text{player})$
   b. $\sigma_{\text{colour = 'gold'} \wedge \text{value = 20}}(\text{item})$
   c. $\pi_{\text{playername}}(\sigma_{\text{itemname = 'Vine Knife'}}(\text{player} \bowtie \text{playeritem} \bowtie \text{item}))$
   d. $\pi_{\text{itemname, value}}(\sigma_{\text{experience = 'Adventurer'} \wedge \text{name != 'coulter'}}(\text{player} \bowtie \text{playeritem} \bowtie \text{item}))$
   e. $\pi_{\text{playername}}(\text{player}) \cap \pi_{\text{itemname}}(\text{item})$

6. a to e. Create the tables yourself, insert some made-up data, and check your answers.

   Because ANY was not covered in labs, a solution is provided. Note that = ANY is exactly the same as writing IN:

   f.
```
SELECT Name
FROM Employee
WHERE DepartmentID = ANY (SELECT DepartmentID
                          FROM Department
                          WHERE DepartmentFloor = 5);
```

7. b, c

8. a. OUTER JOIN is not valid MySQL syntax. Should be LEFT JOIN or RIGHT JOIN
   b. INNER JOIN with no join condition gives cross product. Add ON … clause
   c. BuildingID in SELECT clause is ambiguous. Use Building.BuildingID or Employee.BuildingID
   d. EmployeeName in SELECT clause is not uniquely determined by the GROUP BY column. Reconsider the purpose of the query
   e. HAVING clause doesn't use any aggregate functions. Should be WHERE
   f. GROUP BY must come before ORDER BY. Reorder the clauses
   g. BuildingID in Employee table may be NULL. NOT IN queries do not work correctly when the subquery may include NULLs. Add WHERE BuildingID IS NOT NULL condition to the subquery, or switch to an outer join approach

9. a. i, ii
   b. ii
   c. i, iii
   d. i, iii
   e. None
   f. i, iii

10. a. i: $\frac{7}{160} = 0.0292$; ii: $\frac{1}{40} = 0.025$; iii: $\frac{1}{20} = 0.05$
    b. Cost of best plans: i: 45; ii: 50, iii: 50
    c. Cost of best plans: i: 3; ii: 2, iii: 50

11. I/O cost = 1756, result size = 2000 tuples or 200 pages

12. I/O cost = 250, result size = 1000 tuples or 100 pages

13. a. OrderID → CustomerID, CustomerName: Partial functional dependency
    CustomerID → CustomerName: Transitive functional dependency
    DishID → DishName, DishPrice: Partial functional dependency

    b. Dish (<u>DishID</u>, DishName, DishPrice)
    Order (<u>OrderID</u>, CustomerID, CustomerName)

OrderDish (OrderID[FK], DishID[FK])
Remaining dependency: CustomerID → CustomerName

c. Dish (DishID, DishName, DishPrice)
Order (OrderID, CustomerID[FK])
OrderDish (OrderID[FK], DishID[FK])
Customer (CustomerID, CustomerName)

14. OrderItem (OrderID[FK], ItemID, ItemQuantity)
OrderItemDispatchLocations (OrderID[FK], ItemID[FK], Location)
Order (OrderID, CustomerID[FK])
Customer (CustomerID, CustomerPostcode)

15. a. Team Mascots is a multi-valued attribute (violation of 1NF, 2NF and 3NF)
Team Mascots depends on non-key attribute Team Name (3NF violation)
Team Home Ground depends on non-key attribute Team Name (3NF violation)
Home Ground Location depends on non-key attribute Team Home Ground (3NF violation)

b. Player (PlayerID, PlayerName, PlayerShirtSize, TeamName[FK])
Team (TeamName, TeamHomeGround[FK])
TeamMascot (TeamName[FK], TeamMascot)
TeamHomeGround (TeamHomeGround, Location)

16. a. Now: 49,500 bytes
b. After 5 years: 108,900 bytes

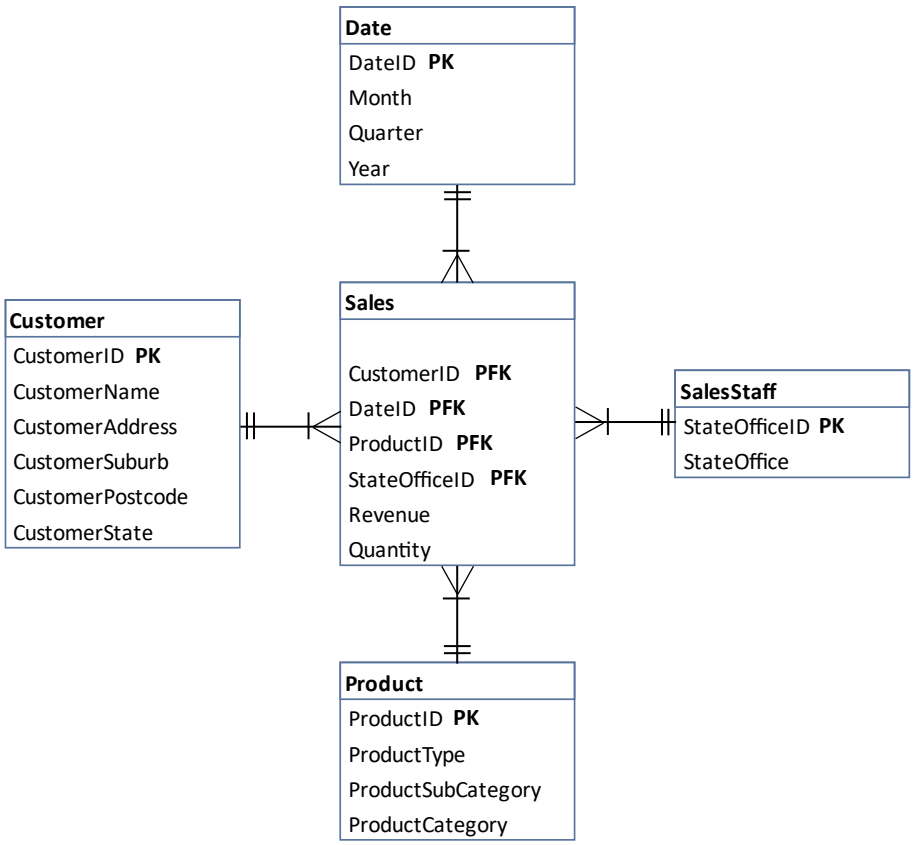
17. Question for discussion

18. Question for discussion

19. a. Facts: Sales revenue, number of products sold
Dimensions: Customer, time, sales office, product

b. Customer: Individual
Time: Month (or finer)
Sales staff: State office
Product: Product type

c. See over page



20. Transactional databases (only): b, c, f, g
    NoSQL databases (only): a, e, i, k
    Both: h, j
    Neither: d