

Student name: Elif Aksoy Student ID: s5417736

*Family name, Given name*

# School of Information and Communication Technology (ICT)

## Database Design

Trimester 2, 2024

**Final Exam**

### Writing time

120 minutes

### Reading time

10 minutes

There are multiple sets of questions. You will be randomly given a set. Please download your set of questions to your computer and work on this document.

Submission consists of one file:

• A Word or PDF document

• You can use draw.io to illustrate your ERD and normalisation step; then attach the picture in this Word document. Alternatively, you can draw on a paper, take picture of that, and then insert into this Word document.

**Question 1 (10 points):**

Given the ERD below, answer the following questions.

A diagram of a software company

Description automatically generated

a) The following query returns 14. **[3 points]**

SELECT COUNT(Apt\_Lvl\_ID)

FROM STORESTAFF

WHERE Apt\_Lvl\_ID <> (

SELECT Apt\_Lvl\_ID

FROM APPOINTMENT

WHERE Apt\_Lvl\_Desc = 'Supervisor'

);

Student A concludes that “Our database contains information on 14 people who are not supervisors”.

***Question***: Do you think the conclusion from A is correct? Provide a clear justification for your reasoning.

Yes, it is correct because the APPOINTMENT table contains the “Supervisor” description that corresponds with Apt\_Lvl\_ID. And the STORESTAFF table contains all staff members and the Apt\_Lvl\_ID reflects there appointment level.

b) The following query attempts to return the max hourly rate of each store. **[3 points]**

*Query 1:*

SELECT Str\_Num,

(SELECT MAX(Hourly\_Rate)

FROM SALARYRATE sr

WHERE sr.Str\_Num = s.Str\_Num) AS Max\_Hourly\_Rate

FROM STORE s;

*Query 2:*

SELECT Str\_Num, Hourly\_Rate as "Max\_Hourly\_Rate"

FROM SALARYRATE

GROUP BY Str\_Num

***Question***: Given any data, will these two queries return the same results? If yes, provide a clear justification for your reasoning. If no, give an example and explanation. Compare the performance of these two queries.

Query 1 retrieves each stores Str\_Num and performs a correlated subquery for each store that finds the MAX(Hourly\_Rate) from the SALARY RATE. Whilst, Query 2 selects the Srr\_Num and the Hourly\_Rate from the SALARY RATE table and groups rows by Str\_Num. But the aggregate function is missing hence the query is incomplete because it does not specify the handling of multiple rows.

No, they will not return the same result because not only is Query 2 written incorrectly but is also missing an aggregate function for the Hourly\_Rate in the GROUP BY line.

c) Given these two queries: **[4 points]**

*Query 1:*

SELECT Str\_Num, MAX(Hourly\_Rate) AS Max\_Hourly\_Rate

FROM SALARYRATE

GROUP BY Str\_Num

HAVING MAX(Hourly\_Rate) >= (SELECT MAX(Hourly\_Rate) FROM SALARYRATE WHERE Str\_Num = SALARYRATE.Str\_Num);

*Query 2:*

SELECT DISTINCT Str\_Num,

(SELECT MAX(Hourly\_Rate)

FROM SALARYRATE sr

WHERE sr.Str\_Num = s.Str\_Num) AS Max\_Hourly\_Rate

FROM SALARYRATE s;

***Question***: Given any data, will these two queries return the same results? If yes, provide a clear justification for your reasoning. If no, give an example and explanation. Compare the performance of these two queries.

Yes, because in the SALARYRATE section, query 1 returns the maximum hourly rate for each store and query 2 does the same.

A further explanation is that in Query 1, the GROUP BY determines each stores maximum hourly rate. And DISTINCT in query 2 ensures that each store appears only once.

**Question 2 (10 points):**

Your company has been engaged by Phones'R'Us, a retail chain specializing in mobile phones, to design a database system for managing customer orders. Your colleague, Colleague A, has proposed the following ERD.

\**Due to time constraints for the exam, additional factors such as color options and storage capacities will not be included in the question.*

A diagram of a product

Description automatically generated

A snapshot of the database from the design of Colleage A:

|  |  |  |
| --- | --- | --- |
| **Product** | | |
| **PhoneTypeID** | **PhoneTypeName** | **WarrantyPeriod** |
| apple16 | Apple iPhone 16 | 2 years |
| apple16pro | Apple iPhone 16 Pro | 2.5 years |
| samsungnote7 | Samsung Galaxy Note 7 | 5 years |
| … | … | … |

|  |  |  |  |
| --- | --- | --- | --- |
| **OrderItem** | | | |
| **OrderItemID** | **OrderID** | **PhoneTypeID** | **Quantity** |
| 1001 | 132 | apple16pro | 2 |
| 1046 | 555 | samsungnote7 | 2 |
| 1064 | 555 | Apple16 | 3 |
| … | … | … |  |

In the ERD above, the "Product" entity in the ERD represents specific phone models, such as iPhone 16, iPhone 16 Pro, Samsung Galaxy Note 7,….

After reviewing the proposed ERD, your Colleague B raised concerns about its ability to track the store that sold each phone and the corresponding customer order numbers, which would be essential if a customer wants to return a product. Colleague B suggests that we should be able to trace this information using the phone's serial number. The below is the version of ERD made by Colleague B.

A diagram of a product

Description automatically generated

A snapshot of the database from the design of Colleage B:

|  |  |  |
| --- | --- | --- |
| **Product** | | |
| **PhoneSerialNumber** | **PhoneTypeName** | **WarrantyPeriod** |
| 350750724344136 | Apple iPhone 16 | 2 years |
| 35-332907 | Google Pixel 9 | 4 years |
| … | … | … |

|  |  |  |
| --- | --- | --- |
| **OrderItem** | | |
| **OrderItemID** | **OrderID** | **PhoneSerialNumber** |
| 1001 | 131 | 350750724344136 |
| 1045 | 190 | 35-332907 |
| 1493 | 131 | 350734231412343 |
| … | … | … |

However, your Colleague C, argues that recording every phone’s serial number during the stock-in process would be impractical, as it would require scanning thousands of phones.

Do you agree with Colleague C’s idea? If yes, draw the ERD representing that idea. The ERD must comply with the Third Normal Form (3NF), ensuring there are no signs of data redundancy. Draw the revised ERD and attach it below.

A black screen with white text

Description automatically generated

**Question 3 (8 points):**

The following is the relation schema of part of the database that XYZ Solution has recently implemented for their client EzyMovie:

**MOVIE**(Movie ID, MovieName, *StudioID*, MovieBudget)

**STUDIO**(StudioID, StudioName, City)

**ACTOR**(ActorID, ActorName, ActorDateOfBirth)

**MOVIEACTOR**(*MovieID*, *ActorID*, OrderOfAppearance)

Using the above schema, write SQL queries that will answer/process the following:

1. List studio names and their locations for those studios that have the letter r in the third position of their names.

**SELECT** StudioName, City

**FROM** STUDIO

**WHERE SUBSTRING** (StudioName, 3, 1) = ‘r’;

1. List the actor names and the movie names they have starred in.

**SELECT** A.ActorName, M.MovieName

**FROM** ACTOR A

**JOIN** MOVIEACTOR MA **ON** A.ActorID = MA.ActorID

**JOIN** MOVIE M **ON** MA.MovieID = M.MovieID;

1. List actor names in order of their appearance in the movie Fast and Not Furious.

**SELECT** A.ActorName

**FROM** ACTOR A

**JOIN** MOVIEACTOR MA **ON** A.ActorID = MA.ActorID

**JOIN** MOVIE M **ON** MA.MovieID = M.MovieID;

**WHERE** M.MovieName = ‘Fast and Not Furious’

**ORDER BY** MA.OrderOfAppearance;

1. List all actor names who have worked for the studio Paramount Pictures.

**SELECT** DISTINCT A.ActorName

**FROM** ACTOR A

**JOIN** MOVIEACTOR MA **ON** A.ActorID = MA.ActorID

**JOIN** MOVIE M **ON** MA.MovieID = M.MovieID;

**JOIN** STUDIO S **ON** M.StudioID = S.StudioID

**WHERE** S.StudioName = ‘Paramount Pictures’;

1. List the total number of movies produced by each studio.

**SELECT** S.StudioName, COUNT(M.MovieID) AS TotalMovies

**FROM** STUDIO S

**JOIN** MOVIE M ON S.StudioID = M.StudioID

**GROUP BY** S.StudioName;

1. List the names of actors who have not appeared in any movie

**ELECT** A.ActorName

**FROM** ACTOR A

**LEFT JOIN** MOVIEACTOR MA **ON** A.ActorID = MA.ActorID

WHERE MA.MovieID is NULL;

1. Find the average budget of movies produced by each studio.

SELECT S.StudioName, AVG(M.MovieBudget) AS AverageBudget

FROM STUDIO S

JOIN MOVIE M ON S.StudioID = M.StudioID

GROUP BY S.StudioName;

1. List all studios located in Burbank and the total number of actors who have worked for these studios.

**SELECT** S.StudioName, COUNT(DISTICNT A.ActorID AS TotalActors

**FROM** STUDIO S

**JOIN** MOVIE M **ON** S.StudioID = M.StudioID

**JOIN** MOVIEACTOR MA **ON** M.MovieID = MA.MovieID

**JOIN** ACTOR A **ON** MA.ActorID = A.ActorID

**WHERE** S.City = ‘Burbank’

**GROUP BY** S.StudioName;

**Question 4 (2 points):**

MediCore Health Services has gathered detailed data on patient visits from the past five years, containing millions of records with attributes like medical history, test results, prescriptions, and treatment outcomes. Due to limited computational resources and the need for faster processing, the analytics team needs to prepare this data for further analysis to identify trends and insights.

Considering the following requirements:

* The team needs to minimize computational costs while ensuring that the most important features of the data are retained.
* The data must be processed in such a way that the analysis can effectively focus on relevant patterns without being overwhelmed by noise or irrelevant information.

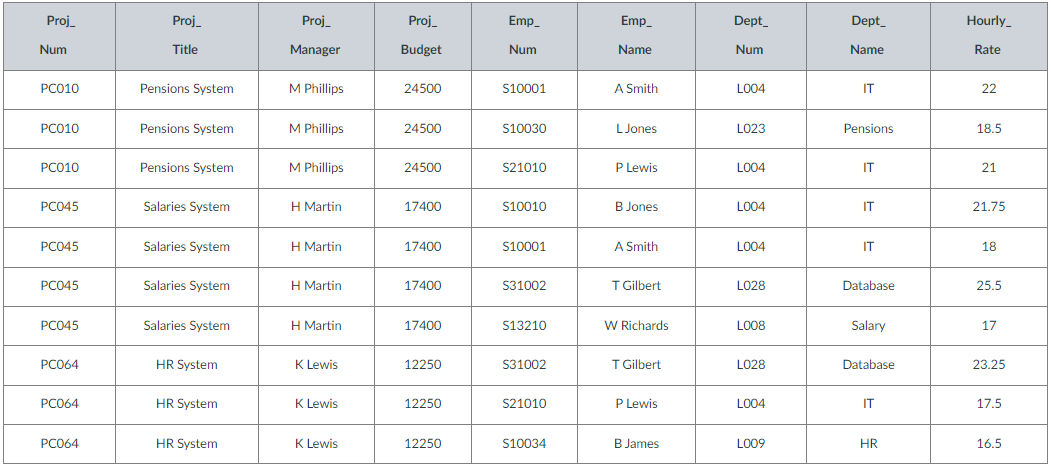
Which approach—data sampling or data reduction—would be most appropriate for preparing this dataset for analysis?

Explain your choice with a clear justification.

Data reduction would be the most appropriate for preparing this data analysis because in data reduction, redundant features are removed while essential patterns are preserved, ensuring valuable insights remain intact. In addition, data reduction techniques such as Principal Component Analysis, or feature selection help remove noises and/or irrelevant information. Making data reduction the best approach.

**Question 5 (5 points):**

Given the following table and functional dependencies:



1. Proj\_Num → Proj\_title, Proj\_Manager, Proj\_Budget

|  |  |  |  |
| --- | --- | --- | --- |
| Proj\_Num (PK) | Proj\_Title | Proj\_Manager | Proj\_Budget |
| PC010 | Pension System | M Phillips | 24500 |
| PC045 | Salaries System | H Martin | 17400 |
| PC065 | HR System | K Lewis | 12250 |

Primary Key; Proj\_Num

1. Emp\_Num → Emp\_Name, Dep\_Num, Dept\_Name

|  |  |  |
| --- | --- | --- |
| Emp\_Num (PK) | Emo\_Name | Dept\_Num (FK) |
| S10001 | A Smith | L004 |
| S10030 | L Jones | L023 |
| S21010 | P Lewis | L004 |
| S10010 | B Jones | L004 |
| S31002 | T Gilbert | L028 |
| S13210 | W Richards | L008 |
| S10034 | B James | L009 |

Primary Key: Emp\_Num

Foreign Key: Dept\_Num (Links to Departments Table)

1. Dept\_Num → Dept\_Name

|  |  |
| --- | --- |
| Dept\_Num (PK) | Dept\_Name |
| L004 | IT |
| L023 | Pensions |
| L028 | Database |
| L008 | Salary |
| L009 | HR |

Primary Key: Dept\_Num

1. {Proj\_Num, Emp\_Num} → Hourly\_Rate

|  |  |  |
| --- | --- | --- |
| Proj\_Num (FK) | Emp\_Num (FK) | Hourly\_Rate |
| PC010 | S10001 | 22 |
| PC010 | S10030 | 18.5 |
| PC010 | S21010 | 21 |
| PC045 | S10010 | 21.75 |
| PC045 | S10001 | 18 |
| PC045 | S31002 | 25.5 |
| PC045 | S13210 | 17 |
| PC064 | S21002 | 23.25 |
| PC064 | S21010 | 17.5 |
| PC064 | S10034 | 16.5 |

Composite Primary Key; Proj\_Num, Emp\_Num

Foreign Keys: Proj\_Num, Emp\_Num

Convert the table into 3NF tables showing all primary keys and foreign keys.

Note that you can either use dependency diagram or functional dependencies for each relation while applying normalisation steps.

**Question 6 (10 points):**

The PharmaCity is a chain of pharmacies, below is the information:

* Pharmacies appoint a supervisor for each contract. There must always be a supervisor for each contract, but the contract supervisor can change over the lifetime of the contract.
* Each pharmacy has a name, address, and phone number.
* Every patient has a primary physician. Every doctor has at least one patient.
* Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.
* Patients are identified by their Medicare numbers, and their names, addresses, and ages must be recorded.
* Doctors are identified by a Practitioner number. For each doctor, the name, specialty, and years of experience must be recorded.
* Each pharmaceutical company is identified by name and has a phone number.
* For each drug, the trade name and formula must be recorded. Each drug is sold by a given pharmaceutical company, and the drug name identifies a drug uniquely. If a pharmaceutical company is deleted, you need not keep track of its products any longer.
* Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors.
* Each prescription has a date and a quantity associated with it. You can assume that, if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored.
* Pharmaceutical companies have long-term contracts with pharmacies. A pharmaceutical company can contract with several pharmacies, and a pharmacy can contract with several pharmaceutical companies. For each contract, you have to store a start date, an end date, and the text of the contract.

According to the information above, draw a corresponding logical ERD using the Crow’s Foot notations. Please note:

* You can decide the names for entities, attributes and relationships, but you should use UPPERCASE to name your entities and use Capitalisation to name attributes.
* Identity the primary key (PK) and foreign key (FK) attributes.
* Clearly indicate connectivity and participation between entities.
* Document all assumptions that you have made. Be realistic and not contradict anything that is in the description.

In particular, in the ERD you need to show:

1. All entities with their attributes, including PK and FK attributes,
2. Relationship (with name) between entities, and
3. Connectivity and participation between entities

A black background with white rectangles

Description automatically generated

*End of questions*