THE UNIVERSITY OF MELBOURNE

School of Computing and Information Systems

INFO20003 Database Systems

Sample exam

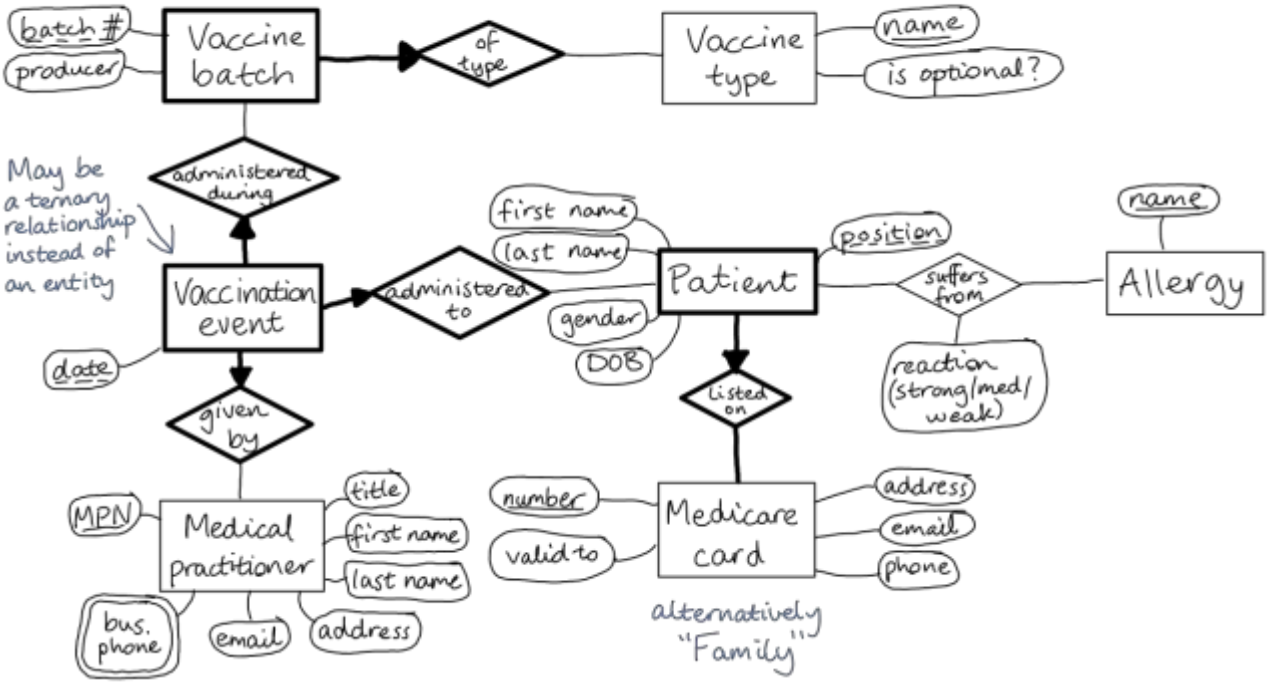
Reading Time: 15 minutes

Writing time: 120 minutes

SUGGESTED SOLUTIONS

Section 1 – ER Modelling

Here's one possible solution in Chen's notation:



Section 2 – SQL-DDL

```
CREATE TABLE TEAM (
  Team_ID      CHAR(4) NOT NULL,
  Team_Name    VARCHAR(25),
  Captain_ID   CHAR(4) NOT NULL,
  PRIMARY KEY (Team_ID),
  FOREIGN KEY (Captain_ID) REFERENCES SWIMMERS (Swimmer_ID)
);

CREATE TABLE SWIMMERS (
  Swimmer_ID      CHAR(4) NOT NULL,
  Swimmer_fname   VARCHAR(25),
  Swimmer_lname   VARCHAR(25),
  Swimmer_email   VARCHAR(35),
  Team_ID         CHAR(4) NOT NULL,
  PRIMARY KEY (Swimmer_ID),
  FOREIGN KEY (Team_ID) REFERENCES TEAM (Team_ID)
);
```

It is not necessary to write NOT NULL when defining the Team_ID and Swimmer_ID columns, as MySQL automatically applies a NOT NULL constraint to primary key columns.

Section 3 – SQL-DML

The relations are repeated here for your convenience.

employees (empid, lastname, firstname, hiredate, address, phone, managerid[FK])

orders (orderid, custid[FK], empid[FK], orderdate, shippeddate, freight, shipname)

customers (custid, companyname, contactname, address, phone)

orderdetails (orderid[FK], productid[FK], quantity, discount)

products (productid, productname, unitprice, discontinued)

Q.3A. Write a query that returns customers (company names) and the details of their orders (orderid and orderdate), including customers who placed no orders.

(3 marks)

```
SELECT C.companyname, O.orderid, O.orderdate
FROM Customers AS C
  LEFT JOIN Orders AS O ON O.custid = C.custid
```

Note: LEFT JOIN and LEFT OUTER JOIN are equivalent; OUTER JOIN (only) is NOT a correct syntax. You may also use NATURAL LEFT JOIN or NATURAL LEFT OUTER JOIN.

Q.3B. Write a query that returns the first name and last name of employees whose manager was hired prior to 01/01/2002.

(4 marks)

```
SELECT E.firstname, E.lastname
FROM Employees AS E
  INNER JOIN Employees AS MNGR ON E.managerid = MNGR.empid
WHERE MNGR.hiredate < '20020101'
```

Note 1: JOIN and INNER JOIN (as keywords) are equivalent.

Note 2: In MySQL, date and time values can be represented in several formats, such as quoted strings or as numbers, depending on the exact type of the value and other factors. For example, in contexts where MySQL expects a date, it interprets any of '2015-07-21', '20150721', and 20150721 as a date. Also for the year part one can use either 'YYYY-MM-DD' or 'YY-MM-DD' format. A “relaxed” syntax is permitted: Any punctuation character may be used as the delimiter between date parts. For example, '2012-12-31', '2012/12/31', '2012^12^31', and '2012@12@31' are equivalent.

Q.3C. Write a query that returns customers (customer ID) whose company name is 'Google', and for each customer return the total number of orders and total quantities for all products that were not discontinued ('1' means discontinued, '0' not discontinued).

(5 marks)

```
SELECT C.custid, COUNT(O.orderid) AS numorders, SUM(OD.quantity) AS totalqty
FROM Customers AS C
  INNER JOIN Orders AS O ON O.custid = C.custid
  INNER JOIN OrderDetails AS OD ON OD.orderid = O.orderid
  INNER JOIN Products AS P ON OD.productid = P.productid
WHERE C.company = 'Google'
  AND P.discontinued = '0'
GROUP BY C.custid;
```

Note 1: JOIN and INNER JOIN are equivalent. You may also use NATURAL JOIN here.

Note 2: Markers were instructed to accept the zero (0) both with/without quotes for `discontinued`.

Q3D. Write a query that returns the ID and company name of customers who placed orders in 2007 but not in 2008.

```
SELECT custid, companyname
FROM Customers
WHERE custid IN
  (SELECT custid
   FROM Orders
   WHERE orderdate >= '20070101'
     AND orderdate < '20080101')
  AND custid NOT IN
  (SELECT custid
   FROM Orders
   WHERE orderdate >= '20080101'
     AND orderdate < '20090101');
```

(8 marks)

Note 1: It is possible to solve this query using EXISTS/NOT EXISTS or = ANY/<> ALL.

Note 2: In MySQL, date and time values can be represented in several formats, such as quoted strings or as numbers, depending on the exact type of the value and other factors. For example, in contexts where MySQL expects a date, it interprets any of '2015-07-21', '20150721', and 20150721 as a date. Also for the year part one can use either 'YYYY-MM-DD' or 'YY-MM-DD' format. A "relaxed" syntax is permitted: Any punctuation character may be used as the delimiter between date parts. For example, '2012-12-31', '2012/12/31', '2012^12^31', and '2012@12@31' are equivalent.

Section 4 – Query Processing – Joins

Q4A. What is the cost (in disk I/O's) of performing this join using the **Block-oriented Nested Loops Join** algorithm? Provide the formulae you use to calculate your cost estimate.

# of blocks = ceil(NPages(outer) / (Number of Buffers – 2)) = ceil(1,000/500) = 2

Total cost = NPages(outer) + (# of blocks × NPages(inner))

= 1,000 + 2 × 50,000 = **101,000** I/O

Q4B. What is the cost (in disk I/O's) of performing this join using the **Hash Join** algorithm? Provide the formulae you use to calculate your cost estimate.

Total cost = 3 × (1,000 + 50,000) = **153,000** I/O

Q4C. In comparing the cost of different algorithms, we count I/O (page accesses) and ignore all other costs. What is the reason behind this approach?

Because the I/O cost dominates the overall cost. One I/O corresponds to millions of CPU cycles.

Note: Any discussion that states that I/O is much more expensive than CPU is acceptable. (*However we haven't focused on CPU cost in this semester, so you can ignore this question.)*

Q4D. Which approach should be the least expensive for the given buffer size of 502 pages:

1. Simple Nested Loops Join
2. Page-oriented Nested Loops Join
3. Block-oriented Nested Loops Join
4. Hash Join

Please write the number of the correct response. No need to provide formulae for question 4D.

Section 5 – Query Processing – Indexing

Q5A. Compute the estimated result size of the query, and the reduction factor of each filter.

$RF_{qty} = (20 - 15)/20 = 0.25$
$RF_{discount} = (100 - 90)/100 = 0.1$
Result size = NTuples(OrderDetails) × RF_{qty} × $RF_{discount}$
= 100,000 × 0.1 × 0.25 = **2,500** tuples

Q5B. Compute the estimated cost of the *best plan* assuming that a *clustered B+ tree* index on *quantity* is (the only index) available. Suppose there are 200 index pages.

Cost of full scan = NPages(OrderDetails) = 100,000/100 = **1000** I/O
Cost of index scan = (NPages(I) + NPages(OrderDetails)) × $RF_{quantity}$
= (200 + 1000) × 0.25 = **300** I/O
The clustered B+ tree index is the best plan here, with a cost of **300** I/O.

Q5C. Compute the estimated cost of the *best plan* assuming that an *unclustered Hash* index on *discount* is (the only index) available.

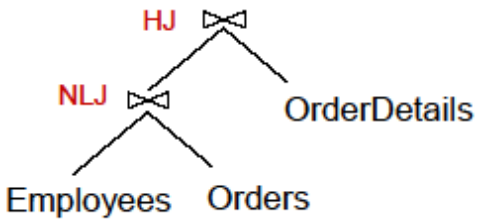
Cost of full scan = NPages(OrderDetails) = 100,000/100 = **1000** I/O
HASH INDEX IS NOT APPLICABLE HERE BECAUSE WE HAVE A RANGE QUERY.
The cost of the best plan here is **1000** I/O.

Q5D. If you are given complete freedom to create one index to speed up this query, which index would be the best one to answer this query? Please give complete information about the index, e.g. is it clustered or unclustered, is it hash or B+-tree, which attributes will it cover.

Clustered B+ tree index on (qty, discount)
OR
Clustered B+ tree index on (discount, qty)

Section 6 – Query Optimisation

Q6A. Compute the cost of the plan shown below. NLJ is a *Page-oriented* Nested Loops Join. Assume that *empid* is the candidate key of Employees, *orderid* is the candidate key of Orders, and 100 tuples of a resulting join between Employees and Orders can fit on one page.



Number of resulting tuples for Employees JOIN Orders = (1/100,000) × (100,000 × 500,000) = 500,000 tuples
Number of pages for Employees JOIN Orders = 500,000 / 100 = 5,000 pages
Cost of scanning Employees = 1,000 I/O
Cost to join with Orders = 1,000 × 5,000 = 5,000,000 I/O
Cost to join with OrderDetails = 2 × 5,000 + 3 × 10,000 = 40,000 I/O

Total cost = 1000 + 5,000,000 + 40,000 = **5,041,000** I/O

Q6B. Would the plan presented below be a valid candidate that System R would consider and compute cost for during query optimisation? Why?

No, cartesian products (cross products) are not considered during query optimization of System R.

Q6C. Consider the query presented below. Does the following equivalence class hold? Yes/No and Why?

$$\Pi_{\text{firstname, lastname}}(\sigma_{\text{quantity}> 5 \wedge \text{freight} <100}(\text{Employees} \bowtie \text{Orders} \bowtie \text{OrderDetails}))$$

$$\leftrightarrow \sigma_{\text{quantity}> 5 \wedge \text{freight} <100}(\Pi_{\text{firstname, lastname}}(\text{Employees} \bowtie \text{Orders} \bowtie \text{OrderDetails}))$$

No, if we project only firstname and lastname, we cannot filter over quantity and freight later on, as these attributes were lost in the projection.

Section 7 – Normalisation

Item ID	Description	Dept	Dept Name	Dept Head	Quan	Cost/Unit	Inventory Value
4011	5 ft desk	MK	Marketing	Jane Thompson	5	200	1000
4020	File cabinet	MK	Marketing	Jane Thompson	10	75	750
4005	Executive chair	MK	Marketing	Jane Thompson	5	100	500
4036	5 ft desk	ENG	Engineering	Ahmad Rashere	7	200	1400

1NF: There are no multivalued attributes, so this table is in 1NF.
2NF: There are no partial functional dependencies, so the table is in 2NF.

3NF: The determinant of the FD:

Dept → Dept Name, Dept Head

is not a key attribute. This transitive FD violates 3NF.
Additionally, another transitive FD (not stated in the question) exists:

Quan, Cost/Unit → Inventory Value

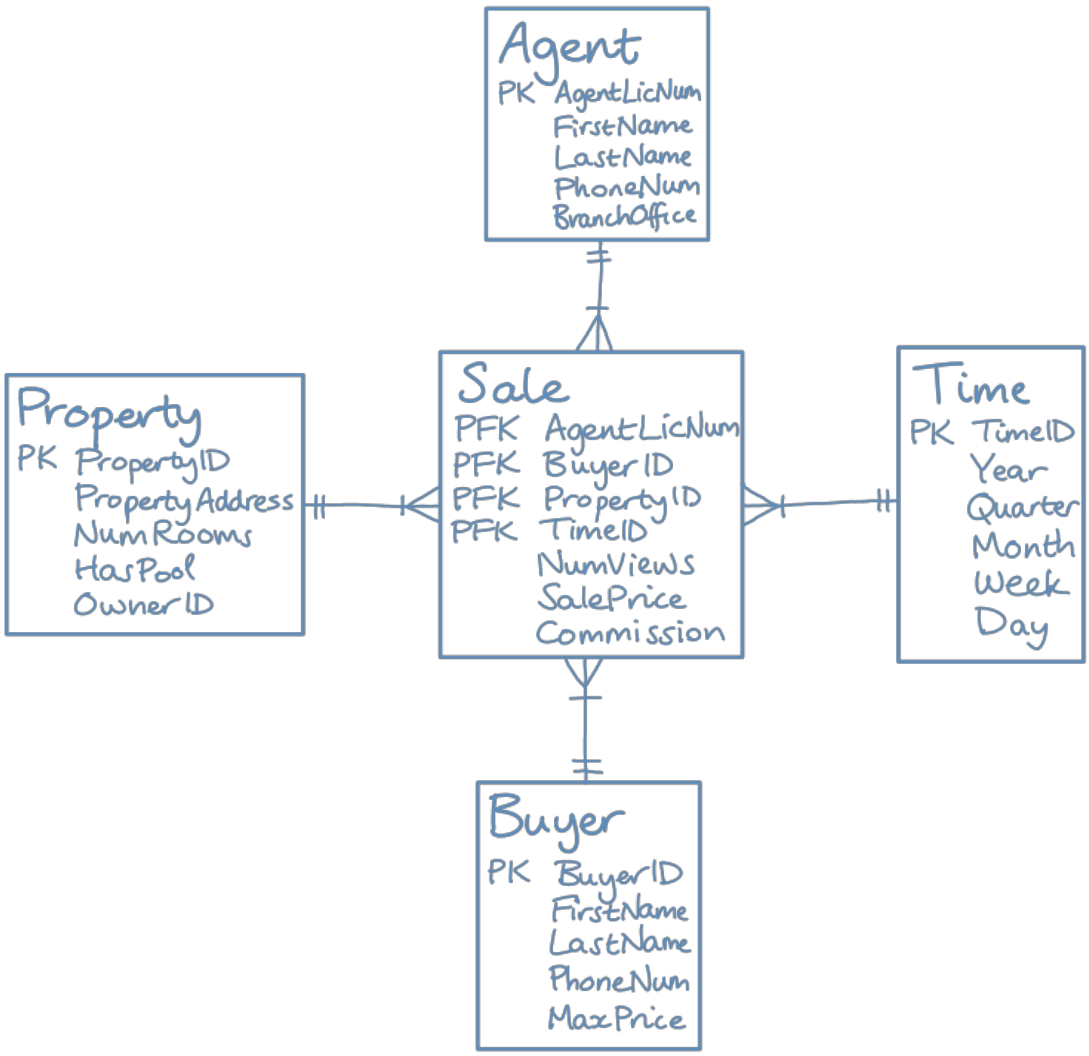
Because there is a simple arithmetic correspondence between these columns (Quan × Cost/Unit = Inventory Value), this FD can be resolved by removing the redundant Inventory Value column.
Since this FD hasn't been stated, both solutions (with and without Inventory value in the Inventory table) were accepted.

The final table structure looks like:

Inventory (Item ID, Description, Dept (FK), Quan, Cost/Unit)
Department (Dept, Dept Name, Dept Head)

Section 8 – Data Warehousing

Q8A. Draw a star schema to support the design of this data warehouse.



Q8B. Why are star schemas preferred over relational database designs to support decision making?

Star schemas are organized around facts (business measures) and dimensions that help with managerial decision making. What's more, they are denormalised, making it faster to aggregate and query data.

Section 9 – NoSQL and Database Administration

Q9A. There are trade-offs between the principles of ACID and BASE. Discuss the trade-off between availability and consistency for relational and NoSQL databases. Illustrate your answer using either the example of Facebook or the example of Twitter.

NoSQL databases need to be available all the time. Facebook (or Twitter) needs to ensure that users can access and post to its site and app all the time, even if the data is not consistent. For example, users expect to always see the number of likes on a post (or retweet count on a tweet); they do not mind if that number is slightly outdated.

Relational databases sacrifice availability for consistency. This means the database needs to be consistent all the time for all the users. If there is potential for inconsistencies (such as two people booking the same seats to watch a movie), the database will sometimes be unavailable.

Q9B. Estimate the disk space requirements **only for the Video table** at go-live and after one month of operation.

First, calculate the row width or row size (in bytes) for each table. Row width is the sum of size of the attribute stored in that row.

- For video table, row size will be 4 + 4 + 20,000 + 8 = 20,016 bytes
- At go live, disk space requirement for Video table will be: 20,016 × 0 = **0** bytes
- After one month, disk space requirement for Video table will be:
  20,016 × 5 × 1,000,000 = $\mathbf{1.10088 \times 10^{11}}$ bytes

Section 10 – Multiple Choice Questions

There are fifteen multiple choice questions. Each question is worth 1 Mark. There is only one correct answer per question. Write the question number and your answer in the script book.

Q10A. Which of the following is **NOT** part of the database development lifecycle?

A) Implementation
B) Maintenance
C) Requirement analysis
D) First-level support

Q10B. A relation which is **NOT** in the conceptual and logical models but is made available to users is a:

A) Data type
B) View
C) Revoke
D) Grant

Q10C. Which of the following is **NOT** a valid join?

A) Sort-merge
B) Hash
C) Nested loop
D) Heap

Q10D. Which of the following is **NOT** a true statement about data and information?

A) Dates and audio are two examples of data
B) Information has been processed so that it increases the audience's knowledge
C) Information is used to create data
D) The term "data" refers to raw facts

Q10E. Which of the following is **NOT** true about normal forms?

A) A table in 1NF must have no multivalued attributes
B) A table in 2NF must have no multivalued attributes
C) A table in 2NF must have no transitive functional dependencies
D) A table in 3NF must have no transitive functional dependencies

Q10F. Which of the following is **NOT** one of the basic operations of Relational Algebra?

A) Set-difference
B) Cross-product
C) Equality
D) Union

Q10G. The structure of a Clustered B+ tree Index does **NOT** contain:

A) Data pages
B) Leaf nodes
C) Internal nodes
D) Root node

Q10H. Which of the following is **NOT** a true statement about projection?

A) Projection can be accomplished by hashing
B) The cost of projection by external merge sort depends on the reduction factor
C) Projection algorithms are designed to remove duplicates from the result set
D) Projection by external merge sort uses the same sorting approach as sort-merge join

Q10I. Which of the following is **NOT** a commonly-used level of lock granularity?

A) Field-level locking
B) Row-level locking
C) Table-level locking
D) Database-level locking

Q10J. Which of the following is **NOT** a valid type of logical database backup?

A) Incremental backup
B) Online backup
C) Onsite backup
D) Offline backup

Q10K. Which of the following is **NOT** a normalization concept?

A) Non-key dependency
B) Partial functional dependency
C) Candidate key
D) Determinants

Q10L. Which of the following is **NOT** a file organisation type?

A) Hash file organisation
B) Index file organisation
C) Heap file organisation
D) Sorted file organisation

Q10M. Which of the following is **NOT** a SQL Command category?

A) DDL
B) DML
C) DCL
D) DSL

Q10N. Which of the following is **NOT** a part of query optimisation?

A) Plan enumeration
B) Costing
C) Building a new index
D) Result size estimation

Q10O. Which of the following is **NOT** true about choosing data types?

A) May help improve query optimisation
B) Help DBMS store and use information efficiently
C) Help minimise storage space
D) May help improve data integrity

END OF THE EXAM