THE UNIVERSITY OF AUCKLAND

SEMESTER ONE 2025
Campus: City

TEST

Database Systems

(Time Allowed: TWO hours)

NOTE:

- Restricted (Closed-book; one A4 double-sided cheat sheet permitted).
- There are 100 marks in this test.
- Write your answers in the answer sheet provided.
  - Please write clearly your Name (better to be consistant with your name on canvas), UPI and ID.
  - Please **completely fill the circles** in the answer sheet as shown:
    A ○ B ○ C ● D ○ E ○
  - Please submit ONLY the answer sheet.

1. (2 marks) Which of the following best describes the Relational Model in database systems?

   A. A model that organizes data into tables (relations) consisting of rows and columns.

   B. A model that organizes data in a hierarchical tree structure.

   C. A model where data is stored as objects, with support for inheritance.

   D. A model designed specifically for managing file-based data storage.

   E. A model used to represent data flows in real-time systems.

2. (2 marks) Which of the following operations is <u>not</u> a basic operation of relational algebra?

   A. Selection σ B. Projection π C. Join $\bowtie$ D. Intersection $\cap$
   E. Aggregation, e.g., SUM, AVG

3. (2 marks) Given the following two tables:

   - `Departments(DeptID, DeptName)`
   - `Employees(EmpID, Name, Salary, DeptID)`

   Which of the following SQL queries retrieves the names of departments where the <u>average salary</u> of employees is greater than $70,000?

   A.
   ```
   SELECT D.DeptName
   FROM Departments D JOIN Employees E ON D.DeptID =
   E.DeptID
   WHERE AVG(E.Salary) > 70000;
   ```

   B.
   ```
   SELECT D.DeptName
   FROM Departments D JOIN Employees E ON D.DeptID =
   E.DeptID
   GROUP BY D.DeptID, D.DeptName
   HAVING AVG(E.Salary) > 70000;
   ```

   C.
   ```
   SELECT DeptName
   FROM Employees
   GROUP BY DeptID
   HAVING AVG(Salary) > 70000;
   ```

D.
```
SELECT D.DeptName
FROM Employees E, Departments D
WHERE E.DeptID = D.DeptID
AND AVG(E.Salary) > 70000
GROUP BY D.DeptName;
```

E.
```
SELECT DeptName
FROM Departments
WHERE EXISTS (SELECT * FROM Employees
WHERE Employees.DeptID = Departments.DeptID
AND AVG(Salary) > 70000);
```

4. (2 marks) Which of the following best describes the purpose of integrity constraints in SQL?
   - A. To improve the performance of SQL queries by optimizing indexes
   - B. To automatically back up data at regular intervals
   - C. To define views for easier data access
   - D. To format the output of SQL queries for reporting
   - E. To ensure that data entered into a database adheres to specified rules and relationships

5. (2 marks) Which of the following statements about SQL authorization is correct?
   - A. The `GRANT` command is used to permanently remove a user's access to a table.
   - B. Only the database administrator (DBA) can use the `GRANT` and `REVOKE` commands.
   - C. The `REVOKE` command is used to give additional permissions to users.
   - D. The `GRANT` command can be used to assign privileges such as `SELECT`, `INSERT`, and `UPDATE` to users.
   - E. Authorization in SQL is automatically handled by the database engine and cannot be controlled manually.

6. (2 marks) Which of the following correctly describes the storage in database systems?

   A. Data is stored only in RAM, and blocks are used to organize cache memory.

   B. Blocks are the smallest units of logical data stored in the CPU registers for query processing.

   C. In the storage hierarchy, blocks are fixed-size units of data transferred between disk and main memory.

   D. The storage hierarchy eliminates the need for organizing data in blocks due to high-speed SSDs.

   E. Blocks are only used in buffer management and have no relation to disk storage.

7. (2 marks) What is the primary purpose of using RAID (Redundant Array of Independent Disks) in database and storage systems?

   A. To compress database files and reduce storage requirements.

   B. To enhance data reliability and/or performance by distributing or replicating data across multiple disks.

   C. To improve security by encrypting data across multiple drives.

   D. To increase storage capacity by combining CPU and disk memory.

   E. To allow parallel execution of SQL queries across multiple databases.

8. (2 marks) Which problem is addressed by log-structured merge tree in a database?

   A. Increased Random Writes

   B. Increased Sequential Writes

   C. Increased Sequential Reads

   D. Increased Random Reads

   E. None of the above

9. (10 marks) You've started a new movie-rating website, and you've been collecting data on reviewers' ratings of various movies. There's not much data yet, but you can still try out some interesting queries. Here's the schema:

   Movie (mID, title, year, director), i.e., there is a movie with ID number mID, a title, a release year, and a director.

   Reviewer (rID, name), i.e., the reviewer with ID number rID has a certain name.

   Rating (rID, mID, stars, ratingDate), i.e., the reviewer rID gave the movie mID a number of stars rating (1-5) on a certain ratingDate.

   (a) (5 marks) Which of the following SQL queries correctly finds the movie(s) with the **highest average rating**, returning their title(s) and average rating?

   A.
```
SELECT title, AVG(stars) AS avg_rating
FROM Movie JOIN Rating USING (mID)
GROUP BY mID, title
HAVING AVG(stars) = (SELECT MAX(AVG(stars))
  FROM Rating
  GROUP BY mID);
```

   B.
```
SELECT title, AVG(stars) AS avg_rating
FROM Movie JOIN Rating USING (mID)
GROUP BY mID, title
ORDER BY AVG(stars)
LIMIT 1;
```

   C.
```
SELECT title, stars
FROM Movie JOIN Rating USING (mID)
WHERE stars = (SELECT MAX(stars) FROM Rating);
```

   D.
```
SELECT mID, title
FROM Movie
WHERE mID IN (
  SELECT mID FROM Rating GROUP BY mID
  HAVING AVG(stars) = (SELECT MAX(stars) FROM Rating));
```

E.
```
SELECT title, x
FROM (SELECT AVG(stars) AS x, r.mID AS mID
  FROM Rating r GROUP BY r.mID
) JOIN Movie USING (mID)
WHERE NOT EXISTS (
  SELECT AVG(stars) AS y
  FROM Rating r2 GROUP BY r2.mID HAVING x < y);
```

(b) (5 marks) Find movies that have an **average rating of 4 or higher**, returning their title and average rating.

A.
```
SELECT title, AVG(stars) AS avg_rating
FROM Movie JOIN Rating USING (mID)
WHERE AVG(stars) >= 4
GROUP BY mID;
```
B.
```
SELECT title, AVG(stars) AS avg_rating
FROM Movie JOIN Rating USING (mID)
GROUP BY mID
HAVING AVG(stars) >= 4;
```
C.
```
SELECT title, AVG(stars) AS avg_rating
FROM Movie JOIN Rating USING (mID)
GROUP BY mID
WHERE AVG(stars) >= 4;
```
D.
```
SELECT title, AVG(stars) AS avg_rating
FROM Movie NATURAL JOIN Rating
GROUP BY mID;
WHERE AVG(stars) >= 4;
```
E. None of the above

10. (4 marks) Which of the following statement(s) about database indexing is (are) **true**?

A. A clustered index can only be built on a primary key.

B. A table can have only one secondary index.

C. A secondary index requires additional I/O to access actual data records.

D. All primary indexes are also secondary indexes.

E. None of the above.

11. (4 marks) Consider the two relations: `A(id)` and `B(id, aid, val)`.

    Evaluate the following query:

    ```
    SELECT A.id, B.val
    FROM A FULL OUTER JOIN B ON A.id = B.aid WHERE B.val IS NULL;
    ```

    Which of the following best describes the result of this query?

    A. It returns only rows from table A with no matching entry in B.

    B. It returns rows from table A that have no match in B **and** rows from table B that have no match in A, but only where B.val is NULL.

    C. It behaves the same as a LEFT JOIN when B.val is NULL.

    D. The query is invalid because FULL OUTER JOIN cannot be filtered on B.val in the WHERE clause.

    E. None of the above.

12. (4 marks) Which of the following statements about Entity-Relationship (ER) diagrams is **true**?

    A. A weak entity always has a primary key of its own independent of any other entity.

    B. A relationship set can have attributes of its own, distinct from those of the participating entities.

    C. Multivalued attributes must always be converted into derived attributes.

    D. Each entity set must participate in at least one relationship.

    E. An ER diagram cannot represent many-to-many relationships directly.

13. (10 marks) A relation called **Student** stores all its 28 tuples in a file. Note that a file is physically a sequence of blocks stored on the disk. Each block can accommodate 3 tuples of **Student**. The stored order of the **Student** tuples are, in their StudentIDs, 80, 54, 9, 93, 58, 109, 110, 107, 51, 81, 42, 44, 123, 97, 25, 116, 65, 71, 6, 89, 49, 101, 23, 91, 66, 62, 17, 84. The database is going to execute an **ORDER BY** StudentID operation on the entire **Student** relation by employing external memory sorting algorithm. The system allocates a buffer pool of 3 frames for sorting this relation. Please answer the following questions.

(a) (2 marks) How many runs are created in the create run phase?
A. 2 B. 3 C. 4 D. 5 E. None of these

(b) (4 marks) How many passes are engaged in merging the runs?
A. 0 B. 1 C. 2 D. 3 E. None of these

(c) (4 marks) How many I/Os are engaged in the entire EM-sorting (final reporting does not cost I/Os)? Hint: please calculate based on your answer to (b).
A. 90 B. 60 C. 40 D. 20 E. None of these

14. (22 marks) Consider a relation R and its B+tree where each leaf node (a block on disk) can hold upto 2 tuples, and each internal node (a block on disk) has a fanout of 4. A total number of 15 tuples are inserted sequentially to the initially empty relation R and the corresponding B+tree is updated accordingly. The keys of the inserted tuples are, sequentially: 31, 19, 15, 72, 11, 78, 69, 29, 63, 75, 4, 5, 49, 99, 27.

(a) (2 marks) After the insertion of the tuple of key __________, the root transits from a leaf node to an internal node.
A. 31 B. 19 C. 72 D. 15 E. None of these

(b) (6 marks) After the insertion of the tuple of key __________, the root splits for the second time and the tree height (the number of nodes on the root-to-leaf path) becomes 3.
A. 69 B. 11 C. 29 D. 78 E. None of these

(c) (6 marks) Please choose the state of the children of the root after inserting the tuple of 4. Note that the nodes at the same level are chained up with $\leftrightarrow$.

A. $|15|19| \leftrightarrow |31|69| \leftrightarrow |72|78|$

B. $|15|19|31| \leftrightarrow |69| \leftrightarrow |72|78|$

C. $|15|19|31| \leftrightarrow |69|72| \leftrightarrow |78|$

D. $|15|19| \leftrightarrow |69| \leftrightarrow |78|$

E. None of the above

(d) (2 marks) A B+tree of 3 levels under the same configuration can hold at least __________ tuples.

A. 2   B. 3   [✓ C. 4]   D. 8   E. 16

(e) (3 marks) If the fanout of the internal node is 100, while a leaf can hold up to 10 tuples, a B+tree of 3 levels can hold at most __________ tuples.

A. 10^3 B. 10^4 [✓ C. 10^5]   D. 5×10^5 E. 10^6

(f) (3 marks) Why does batched (bulk) construction of a B$^+$-tree improve I/O performance compared to inserting records one at a time?

A. It avoids using any disk I/O by keeping the tree entirely in memory.

B. It guarantees a perfectly balanced B$^+$-tree, which insertion-based methods cannot achieve.

C. It eliminates the need for sorting records before inserting into the tree.

D. It minimizes the height of the tree by using fewer internal nodes.

[✓ E. It reduces random I/O by writing nodes to disk in large, sequential blocks.]

15. (24 marks) Consider an extendible hashing structure such that 1) each bucket can hold up to two records, 2) a new extendible hashing structure is initialized with g = 0 and one empty bucket, and 3) if multiple keys are provided in a question, assume they are inserted one after the other from left to right.

key	h(key)	key	h(key)
0	0000000000	13	1011000000
1	1000000000	17	1000100000
2	0100000000	27	1101100000
3	1100000000	32	0000010000
5	1010000000	43	1101010000
8	0001000000	64	0000001000
9	1001000000	128	0000000100
11	1101000000	512	0000000001

(a) Starting from an empty table, insert keys 1, 2.

  i. (2 marks) What is the global depth of the resulting table?
  [✓ A. 0] B. 1 C. 2 D. 3 E. None of the above

  ii. (2 marks) What is the local depth of the bucket containing 2?
  [✓ A. 0] B. 1 C. 2 D. 3 E. None of the above

(b) Starting from the result in (a), you insert keys 9, 11.

  i. (2 marks) What is the global depth of the resulting table?
  A. 0 B. 1 [✓ C. 2] D. 3 E. None of the above

  ii. (2 marks) What are the local depths of the buckets for each key?

  A. 1 (Depth 1), 2 (Depth 1), 9 (Depth 1), 11 (Depth 1)
  B. 1 (Depth 3), 2 (Depth 1), 9 (Depth 3), 11 (Depth 3)
  C. 1 (Depth 3), 2 (Depth 1), 9 (Depth 3), 11 (Depth 2)
  [✓ D. 1 (Depth 2), 2 (Depth 1), 9 (Depth 2), 11 (Depth 2)]
  E. None of the above

(c) Starting from the result in (b), you insert keys 13, 27.

  i. (2 marks) What is the global depth of the resulting table?
  A. 1 B. 2 [✓ C. 3] D. 4 E. None of the above

  ii. (2 marks) What are the local depths of the buckets for each new key?

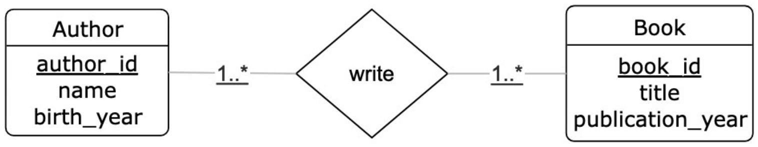
  [✓ A. 13 (Depth 3), 27 (Depth 2)]
  B. 13 (Depth 1), 27 (Depth 2)
  C. 13 (Depth 2), 27 (Depth 2)
  D. 13 (Depth 3), 27 (Depth 3)
  E. None of the above

(d) (4 marks) Starting from (c)'s result, which key(s), if inserted next, will not cause a split?
[✓ A. 5] B. 17 C. 43 [✓ D. 8] E. None of the above

(e) (4 marks) Starting from the result in (c), which key(s), if inserted next, will cause a split and increase the table's global depth?
A. 0 [✓ B. 3] C. 5 [✓ D. 17] E. None of the above

(f) (4 marks) Starting from an empty table, insert keys 32, 64, 128, 512. What is the global depth of the resulting table?
A. 5 B. 6 [✓ C. 7] D. 8 E. ≥ 9

16. (6 marks) Consider an ER diagram with Entities Author and Book. The relationship writes: Each Author can write multiple Books (at least one), and each Book is written by at least one Author.



Given the ER diagram, its corresponding SQL could be __________.

A.
```
CREATE TABLE Author (
    author_id INT PRIMARY KEY,
    name VARCHAR(100), birth_year INT);

CREATE TABLE Book (
    book_id INT PRIMARY KEY,
    title VARCHAR(100),
    publication_year INT, author_id INT,
    FOREIGN KEY (author_id) REFERENCES Author(author_id));
```

B.
```
CREATE TABLE Author (author_id INT PRIMARY KEY,
    name VARCHAR(100), birth_year INT);

CREATE TABLE Book (book_id INT PRIMARY KEY,
    title VARCHAR(100), publication_year INT);

CREATE TABLE Writes (
    book_id INT, author_id INT,
    FOREIGN KEY (book_id) REFERENCES Book(book_id),
    FOREIGN KEY (author_id) REFERENCES Author(author_id));
```

C.
```
CREATE TABLE Author (
    author_id INT PRIMARY KEY,
    name VARCHAR(100), birth_year INT,
    FOREIGN KEY (book_id) REFERENCES Book(book_id));

CREATE TABLE Book (
    book_id INT PRIMARY KEY,
    title VARCHAR(100), publication_year INT);
```

D.
```
CREATE TABLE Author (
    author_id INT PRIMARY KEY,
    name VARCHAR(100), birth_year INT);

CREATE TABLE Book (
    book_id INT PRIMARY KEY,
    title VARCHAR(100), publication_year INT);

CREATE TABLE Writes (
    author_id INT, book_id INT,
    PRIMARY KEY (author_id, book_id),
    FOREIGN KEY (author_id) REFERENCES Author(author_id),
    FOREIGN KEY (book_id) REFERENCES Book(book_id)
);
```

E. None of the above

*************** The end of the problem sheet ****************