INDENG 215 Final Exam

Analysis and Design of Databases

Instructions: Answer all questions. Provide clear and concise explanations for all your answers, including any assumptions you make. Submit your solutions on GradeScope by 6pm PT on December 18; no late work will be accepted. 38 points are possible on this exam. You must sign and submit also the honor code below as part of your submission.

If at all possible, please write your answers on the exam paper (whether electronically or printed), maintaining the spacing and alignment of the questions as written here, which streamlines the grading process significantly. Thank you! :)

Honor Code: As a takehome exam, you may make use of all course materials freely. However, you **may not** discuss the exam or share work *in any capacity* with anyone else, whether electronic or in person, whether they are a student in the course or not. All answers must be your own and be written by yourself, and AI tools may not be used for this exam.

I acknowledge and agree to follow the above:

Signature: Weihang Ding

Name: Weihang Ding

Student ID: 3041911265

I. True / False (9 Points), 1 each

Answer true or false by filling in the corresponding box.

1. In a one-to-Many (1:N) relationship between two entities (e.g., `Student` and `Course`), the standard relational mapping requires creating a separate table containing the primary keys of both entities. T □ F ■

2. You have a table `Employees(EmpID, Name, ManagerID)`. You want to find the names of employees who have the same name as their manager. The following query is valid SQL:

```
SELECT E1.Name
FROM Employees E1 JOIN Employees E2
ON E1.ManagerID = E2.EmpID
WHERE E1.Name = E2.Name
```

T ■ F □

3. In an ER diagram, if `Course` is a Weak Entity dependent on `Department`, and `Course` has a partial key `CourseNum`, then the Primary Key of the mapped relational table `Course` will be the composite (DepartmentID, CourseNum). T ■ F □

4. In SQL three-valued-logic, a statement (UNKNOWN AND FALSE) evaluates to FALSE.
T ■ F □

5. Denormalization helps write-heavy applications (high `INSERT`/`UPDATE` volume) by reducing the need to update multiple tables.
T □ F ■

6. The Isolation property in ACID ensures that the intermediate state of a transaction is invisible to other concurrent transactionsuntil the transaction commits.
T ■ F □

7. In an OLAP system with a Star Schema, the central Fact Table typically stores descriptive text data (e.g., ”Product Name”, ”Store Address”), while the Dimension tables store the numerical metrics (e.g., ”Revenue”, ”Units Sold”).
T □ F ■

8. If a relation has a single-attribute primary key, it is automatically in 1NF.

T □ F ■

9. A transitive dependency occurs when we have a composite primary key, and one of its attributes fully determines another attribute in the relation.

T □ F ■

II. Multiple Choice (5 Points, 1 each)

Select the single best answer for each question by filling in the corresponding box.

1. You are writing a query to find the average salary of employees in each department, but you only want to see departments where the average salary is greater than $80,000.

Which SQL clause would be best to use to perform this filtering?

a) □ `WHERE AVG(Salary) > 80000`

b) ■ `HAVING AVG(Salary) > 80000`

c) □ `GROUP BY AVG(Salary) > 80000`

d) □ `ORDER BY AVG(Salary)`

2. Consider:

Orders(OrderID, CustomerID, CustomerCity, OrderDate)

Primary Key: `OrderID`.

FDs: OrderID → CustomerID, OrderID → OrderDate, and CustomerID → CustomerCity.

What is the highest normal form this relation satisfies?

a) □ 1NF only (Violates 2NF).

b) ■ 2NF (Violates 3NF due to transitive dependency).

c) □ 3NF (It is fully normalized).

3. A university database tracks courses. A single field in the `Course` table is named `Prerequisites`. For a specific row, the data in this field looks like: `"Math101, Phys105, Eng202"`.

This design violates First Normal Form (1NF) primarily because:

a) ■ The values are not atomic; a repeating group is stored in a single column.

b) □ The field depends on the Course ID, which is a transitive dependency.

c) □ Strings cannot contain commas in a relational database.

d) □ There is no primary key defined for the string.

4. You are designing a database for a high-traffic concert ticketing system. When tickets go on sale, thousands of users try to buy the same seat simultaneously.

Which ACID property is most critical to ensure that a single seat is not sold to two different people?

a) □ Durability

b) □ Atomicity

c) □ Consistency

d) ☒ Isolation

5. Consider a table `Inventory(PartID, WarehouseID, Quantity, WarehouseAddress)`. The primary key is `(PartID, WarehouseID)`.

The attribute `WarehouseAddress` depends only on `WarehouseID`, not on `PartID`. This is an example of:

a) □ Transitive Dependency

b) ☒ Partial Dependency

c) □ Cyclic Dependency

d) □ Join Dependency

III. Short Answer

1. **(1pt)** Briefly explain why OLTP (Online Transaction Processing) systems prioritize normalization, whereas OLAP (Online Analytical Processing) systems often allow denormalization.

OLTP systems priortize normalization to minimize redundancy ensuring fast, consistent writes under many concurrent transactions.

OLAP systems often allow denormalization to reduce joins and speed up heavy queries

2. **(3pts; 1pt each)** For each task described below, determine whether each task is best suited for an OLTP or OLAP system, and explain briefly why.

   (a) A warehouse manager needs to check the current level for a specific product code (SKU = 4712) at a distribution center to confirm whether an order can be fulfilled.

   OLTP

   a real-time look up of the current inventory

(b) A regional sales director requests a report that calculates the total revenue generated by all sales representatives in the Northeast region for the entire last fiscal quarter, grouped by product category.

OLAP

a historical, aggregated reporting query

(c) A finance team is running a query to compare the average profit margin of products sold during the current year against the sales data from the previous five years.

OLAP

read-heavy and aggregation focused.

3. **(1pt)** Describe what is meant by an ”Update Anomaly.” Provide a concrete example using a table `StudentAdvisor(Student, Advisor, AdvisorEmail)` where the primary key is `Student`.

An update anomaly happens when the same real-world fact is stored redundantly in multiple rows, changing it requires multiple updates and missing one creates inconsistent data.

Example: if many students have the same advisor updating advisor email for only one student but not others leaves conflicting emails for the same advisor in the table

4. **(1pt)** In the ETL process, why is a ”Staging Area” often used? Why do we not simply load data directly from the source application into the final Data Warehouse tables in one step?

A staging area is a temporary landing zone to validate clean and transform raw source data before it enters the data warehouse.

Reason: sources can be inconsistent or dirty and staging enables reliable retries without impacting the final tables

5. **(6pts)** Consider the following schema for managing a supply chain logistics system:

```
Shipment(Shipment_ID, Warehouse_ID, Warehouse_Location,
    Product_ID, Product_Name, Supplier_ID, Quantity,
    Shipment_Date, In_Stock_at_Warehouse, Memo)
```

Where `Memo` is a note for the shipment, and `In_Stock_at_Warehouse` is true or false depending on whether the product with ID `ProductID` is in stock at the warehouse with ID `WarehouseID`.

(a) **(2pts)** Describe two data anomalies that could occur if one were to work with this schema and explain why they might be problematic.

Update Anomaly: Warehouse_Location is repeated for every shipment from the same warehouse, changing a location requires updating numerous rows.

Deletion Anomaly: if only shipment for a specific product is deleted some information for that product might lost forever.

(b) **(2pts)** Make a list of functional dependencies that should hold for this scenario. If you feel uncertain about any of them, explain your assumptions clearly.

Shipment_ID → Warehouse-ID, Product-ID, Supplier-ID, Quantity, Shipment_Date, Memo

Warehouse-ID → Warehouse_Location

Product_ID → Product_Nane

(Product-ID, Warehouse_ID) → In_Stock_at_Warehouse.

(c) **(2pts)** Using these functional dependencies, decompose this relation into a schema in which all relations are in 3NF, and indicate PKs and FKs in all your relations.

Warehouse ($\underline{\text{Warehouse\_ID}}$, Warehouse_Location)

Product ($\underline{\text{Product\_ID}}$, Product_Name)

Inventory ($\underline{\text{Warehouse\_ID}^F}$, $\underline{\text{Product\_ID}^F}$, In-stock_at_Warehouse)

Shipment ($\underline{\text{Shipment\_ID}}$, Warehouse_ID^F , Product_ID^F , Quantity , Shipment_Date, Memo)

6. **(6pts)** An auto insurance company suspects that an employee may have been involved in an insurance fraud scheme. The company maintains the following database schema for tracking claims and employees:

Schema:

```
Customer(Customer_ID, Name, Email, Phone, Address)
Claim(Claim_ID, Customer_ID, Date_Filed, Claim_Amount, Claim_Status,
        Adjuster_ID, Vehicle_ID)
Employee(Employee_ID, Name, Position)
Vehicle(Vehicle_ID, License_Plate, Make, Model, Year)
Repair(Repair_ID, Claim_ID, Date_Repaired, Repair_Cost, Repair_Shop,
        Adjuster_ID)
```

Additional Information:

- Each claim is assigned to an adjuster (an employee responsible for investigating and approving the claim).
- Repair costs and shops are recorded for each claim where applicable.
- Claims are filed by customers for damages to their vehicles.

(a) **(2pts)** The company suspects that employee 90210 is working as an insider in a fraud scheme. Write a query to identify all claims where:

- The claim amount exceeds $50,000.
- The repair cost associated with the claim is less than 10% of the claim amount.
- The claim was approved by the adjuster with employee ID 90210.

```
SELECT    c.*
FROM    Claim c
JOIN    Repair r    on    c.Claim_ID
= r.Claim_ID
WHERE    c.Claim_Amount > 50000
    And    c.Adjuster_ID= 90210
GROUP BY    c.claim_ID, c.Claim_Amount
Having SUM(r.Repair_Cost) < 0.1 * c.Claim_
                                   Amount.
```

After running this first query, the team is quite certain that there is a fraud scheme happening and aims to identify other parties involved therein.

(b) **(2pts)** Write a query to find the Make and Model of vehicles involved in 4 or more claims in 2024.

```
SELECT v.Make, v.Model
  FROM Vehicle v
  JOIN Claim c  ON v.vehicle_ID = c.Vehicle_ID
  WHERE  c.Date_Filed >= '2024-01-01'
     AND c.Date_Filed = '2025-01-01'
  GROUP BY v.vehicle_ID, V_Make, V.Model
   HAVING COUNT(DISTINCT(c.claim_ID))>=4
```

(c) **(2pts)**Write a query that gives a list of the repair shops at which repairs for claims approved by employee 90210 were made, along with the total repair costs of all repairs associated with those claims for each of those shops. (in case this is unclear, query to answer the questions: At which repair shops are repairs for employee 90210's claims made, and how much $ worth of repairs have this employee's claims given to each of these shops?)

```
SELECT r.Repair_Shop, SUM(r.Repair_COST AS
Total_Repair_Cost
 FROM Repair r
 JOIN  Claim c  ON r.claim_ID = c.Claim_ID
 WHERE  c.Adjuster_ID = 90210
 GROUP BY r.Repair_Shop
```

7. (5 pts)

You are the Lead Data Architect for a new ride-sharing platform. You need to design both the operational database (OLTP) to run the app and the data warehouse (OLAP) for business analytics.

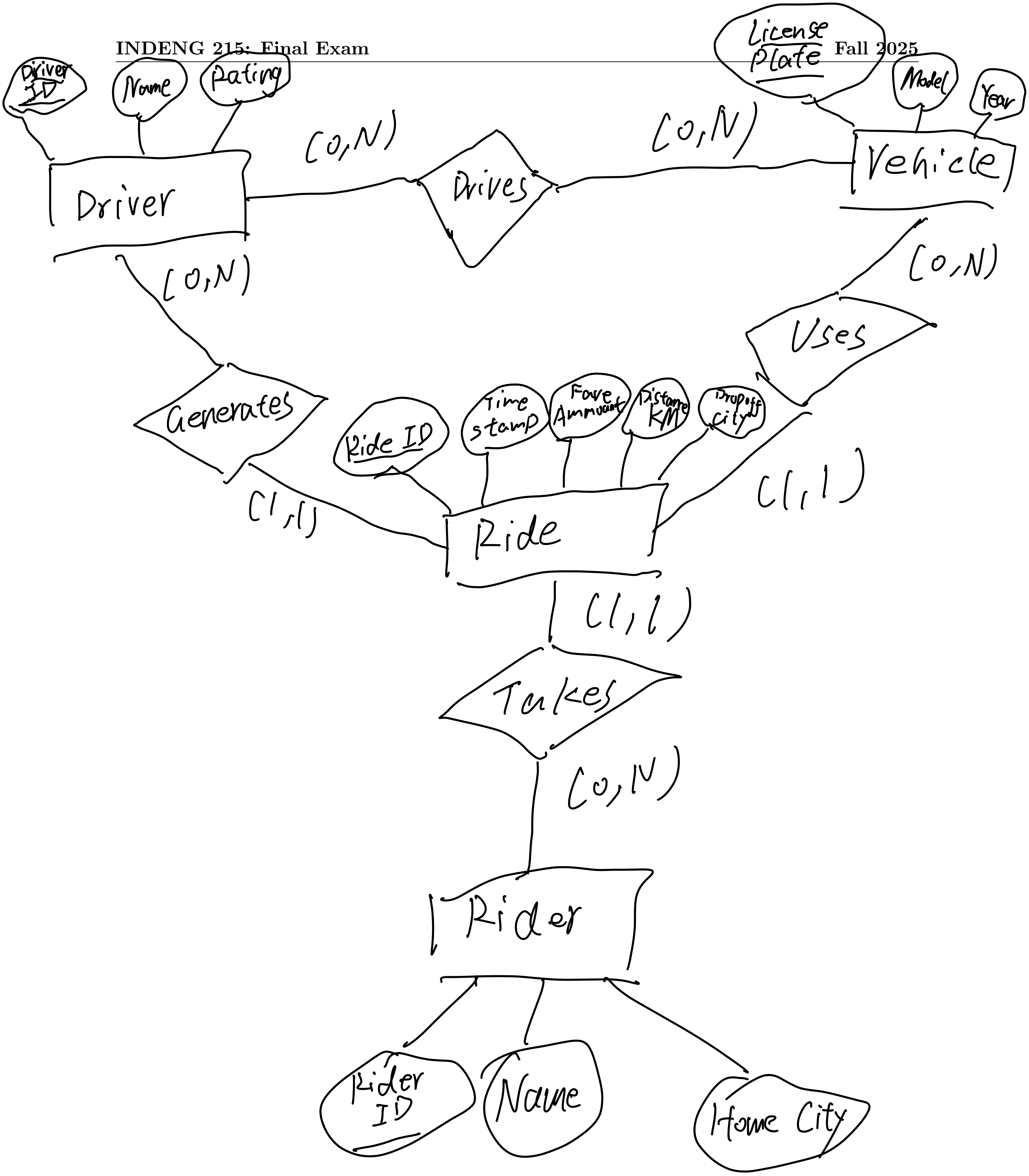
Scenario Requirements:

- The platform tracks Drivers. For each driver, we record a unique `DriverID`, `Name`, and `Rating`.
- Drivers drive Vehicles. A vehicle is identified by its `LicensePlate`, and we also track the `Model` and `Year`. A driver can drive many different vehicles over time, and a vehicle can be driven by multiple drivers (but not at the same time).
- The platform tracks Riders (passengers). For each rider, we record a unique `RiderID`, `Name`, and `HomeCity`.
- A Ride is a trip taken by a specific Rider, driven by a specific Driver, in a specific Vehicle.
- For every Ride, we track a unique `RideID`, the `Timestamp` it started, the `FareAmount`, the `DistanceKM`, and the `DropoffCity`.

(a) **(2pts)** Draw an Entity-Relationship (ER) Diagram for the operational (OLTP) system based on the requirements above. Clearly indicate:

- Entities and their attributes (underline Primary Keys).
- Relationships between entities.
- Cardinality constraints (e.g., 1:N, N:N) for every relationship.

(Draw your diagram in the space provided on the next page.)


Driver ID
Name
Rating
Driver
(0,N)
Drives
(0,N)
License Plate
Model
Year
Vehicle
(0,N)
(0,N)
Uses
Generates
Ride ID
Time stamp
Fare Ammount
Distance KM
Dropoff City
(1,1)
Ride
(1,1)
(1,1)
Takes
(0,N)
Rider
Rider ID
Name
Home City

(b) **(2pts)**
Management wants to analyze the business performance. They specifically want to run queries like:

> "What is the total revenue and average distance per ride, broken down by Vehicle Model, by Dropoff City, and by Month?"

Design a Star Schema to support this analysis. Make sure to clearly:

- Identify your central Fact Table and list its specific Measures (numerical facts) and Foreign Keys.
- Identify at least 3 Dimension Tables. For each dimension, list the table name and 2-3 specific attributes you would suggest to include based on the scenario above (these attributes need not necessarily be present in your ER diagram from part a)).

Central Fact Table: Fact_Rides

FKs: Vehicle_key, Location_Key, Time_Key, Rider_key

Measures: Fare Amount, Distance KM, Ride-Count,

Dimension Tables:

Dim-Vehicle: [Vehicle_key, License Plate, Model, Year]

Dim_Location: [Location_Key. Dropoff City, State),

Dim_Time: (Time_Key, Date, Month, Year)

(c) **(1pt) Analysis**
Briefly explain why using the Star Schema (from Part b) is more efficient for queries such as the example given above than running it directly against the normalized ER schema (from Part a).

The star Schema is more efficient for analytical queries because it reduces complex joins by denormalization data.

8. (1pt) (Team Member Participation / Contributions Rating; full points given for answering) Divide up 100 hypothetical points total among your other team members for their contributions and participation in your team projects (higher points for more contributions / participation, less points for less contributions / participation). Please take into account that sometimes if responsibilities are divided up, some responsibilities inevitably take less time or effort than others. You may also simply answer "all equal" if you feel that your team members all contributed sufficiently.

All equal