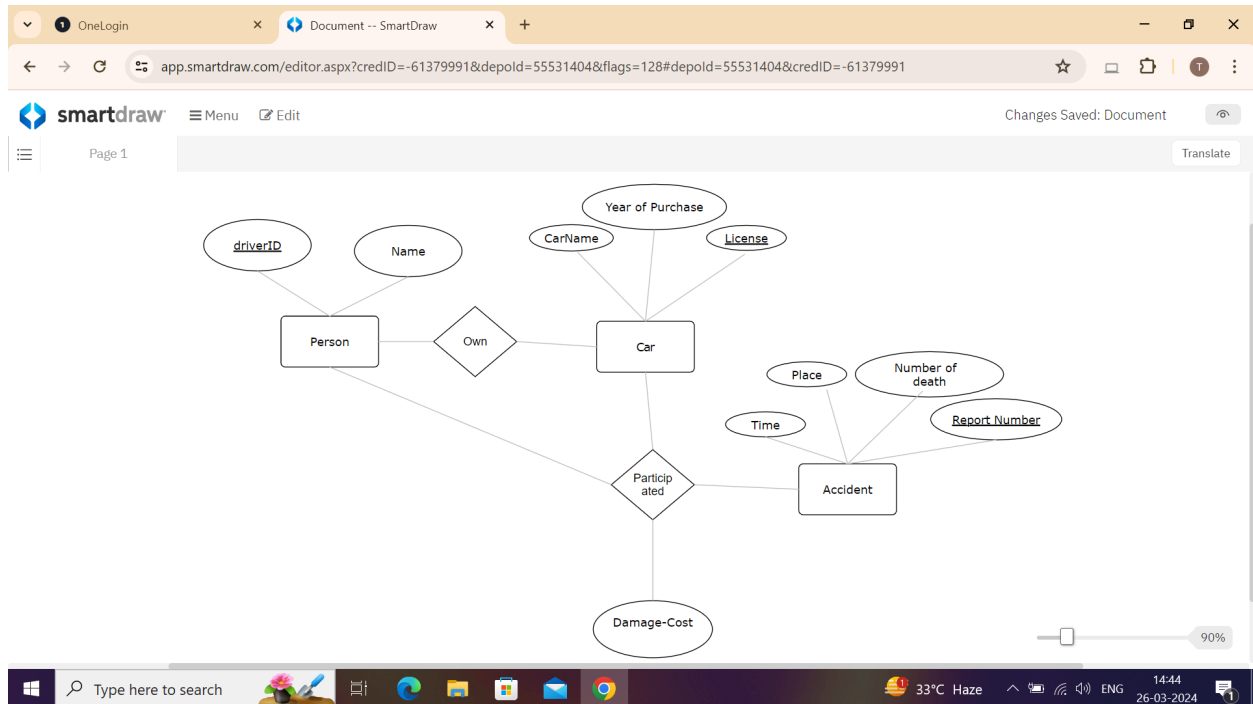


RDBMS ASSIGNMENT

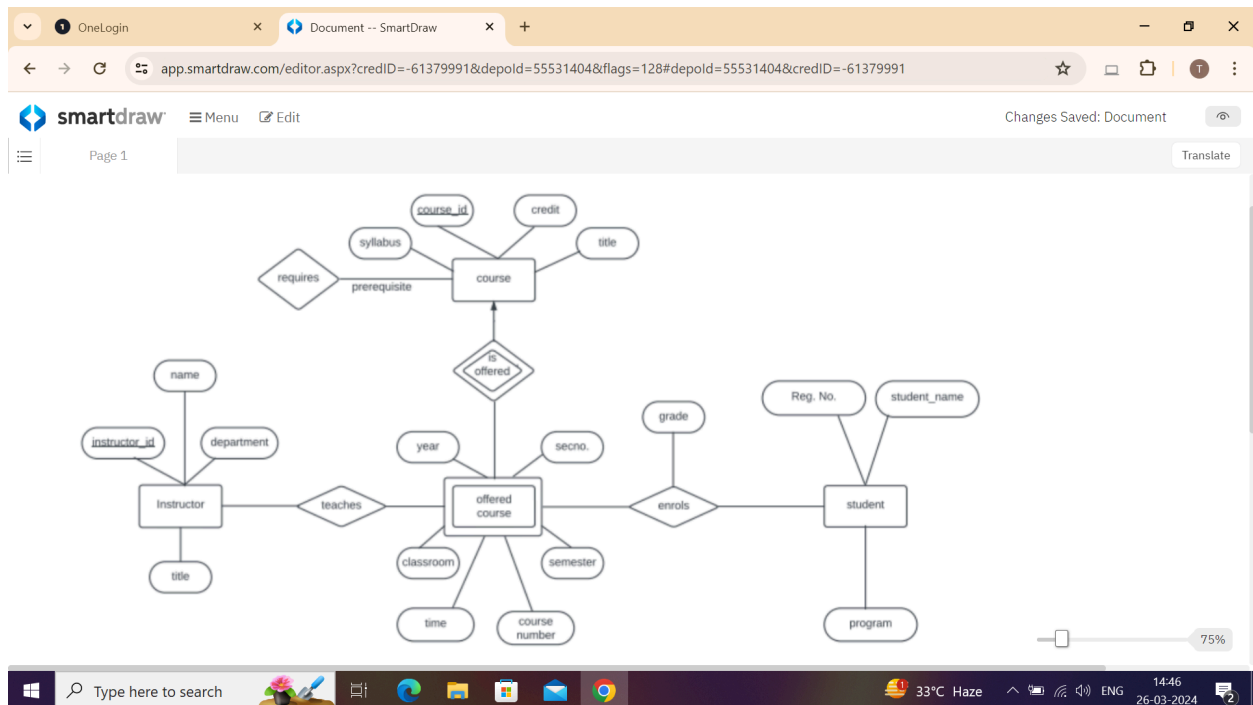
NAME: TANISHQ JAIN

INTERN ID: INT1581

1. Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.



2. A university registrar's office maintains data about the following entities: (a) courses, including number, title, credits, syllabus, and prerequisites; (b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom; (c) students, including student-id, name, and program; and (d) instructors, including identification number, name, department, and title. Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraint.



3. Consider an E-R diagram in which the same entity set appears several times. Why is allowing this redundancy a bad practice that one should avoid whenever possible?

In an Entity-Relationship Diagram (ERD), entity sets correspond to tables in a database. If an ERD contains multiple instances of the same entity set, it suggests that the corresponding table may appear multiple times in the database. However, this repetition of entity sets can lead to missing relationships in the model.

4. Explain a real-time RDBMS system and how it is constructed , detailing using the concepts of RDBMS Primary Key, Secondary Key, Composite Key, Unique constraint, Referential Integrity Constraint, Not Null constraint.

A real-time RDBMS is a system designed to store, manage, and retrieve data in a structured manner. In RDBMS at the top level, there's usually a database, which is a collection of related data tables. Each table in the database consists of rows and columns. Columns represent attributes and rows represent individual records.

a) **Primary Key:** A primary key is a unique identifier for each row in a table. It ensures that each row in the table is uniquely identifiable. Primary keys must have unique values and cannot contain NULL values.

b) **Secondary Key:** Secondary keys are indexes created on columns other than the primary key to improve query performance. They help speed up the retrieval of data based on criteria other than the primary key. Example: In the table of employees, a secondary key might be created on the employee's last name for efficient searching.

c) **Composite Key:** A composite key is a combination of two or more columns that uniquely identify each row in a table. Unlike a primary key, individual columns in a composite key might not be unique by themselves, but their combination is unique. Composite keys are useful when a single column cannot uniquely identify rows in a table. Example: In a table of orders, a composite key might be formed by combining the order ID and the product ID, ensuring uniqueness for each order-product combination.

d) **Unique Constraint:** A unique constraint ensures that the values in a column or a group of columns are unique across all rows in the table. Example: A table of email addresses might have a unique constraint on the email column to prevent duplication of email addresses.

e) **Referential Integrity Constraint:** Referential integrity ensures the validity of data by enforcing relationships between tables. It is typically enforced through foreign key constraints. Foreign keys link a column or group of columns in one table to the primary key or unique key in another table. They ensure that values in the foreign key column(s) match values in the referenced primary or unique key column(s) in another table. Example: In a database with tables for orders and customers, the order table might have a foreign key constraint referencing the customer ID column in the customer table.

f) **Not Null Constraint:** A not null constraint ensures that a column does not contain NULL values. It ensures data integrity. If a column is defined as not null, every row in the table must have a value for that column. Example: In a table of addresses, the column for the city might be defined as not null to ensure every address record includes a city name.

5. Assume a Table product with the following fields (ProductNo (pk), Product_name, Cost_price, Selling_price, Quantity)

Write a SQL query for :

a) Display the product details whose selling_price is double the cost_price

b) Display the product details whose product_name starts with G and quantity is more than 200 .

a) Display the product details whose selling price is double the cost price

```
SELECT * FROM product  
WHERE Selling_price = 2 * Cost_price;
```

b) Display the product details whose product_name starts with G and quantity is more than 200

```
SELECT * FROM product  
WHERE Product_name LIKE 'G%' AND Quantity > 200;
```

6. The room booking side of a small hotel is to be computerized. The hotel has a number of rooms. Each room has a basic (double) price and a supplementary price for extra children. These prices also depend on the time of year- it is more expensive at Christmas, during the summer and around bank holidays, for example. There are 3 seasonal bands. The system must enable the hotel proprietor to answer phone calls from prospective clients (for example, rooms available now and in the future, with costs), make provisional bookings, do mailings of previous clients, prepare clients' bill (ignore extras such as papers, drinks etc). Recognise various entities and relationship among them. Construct an E-R diagram for the above.

Editing is disabled. To edit, click the Edit button. [Edit this Document.](#)

