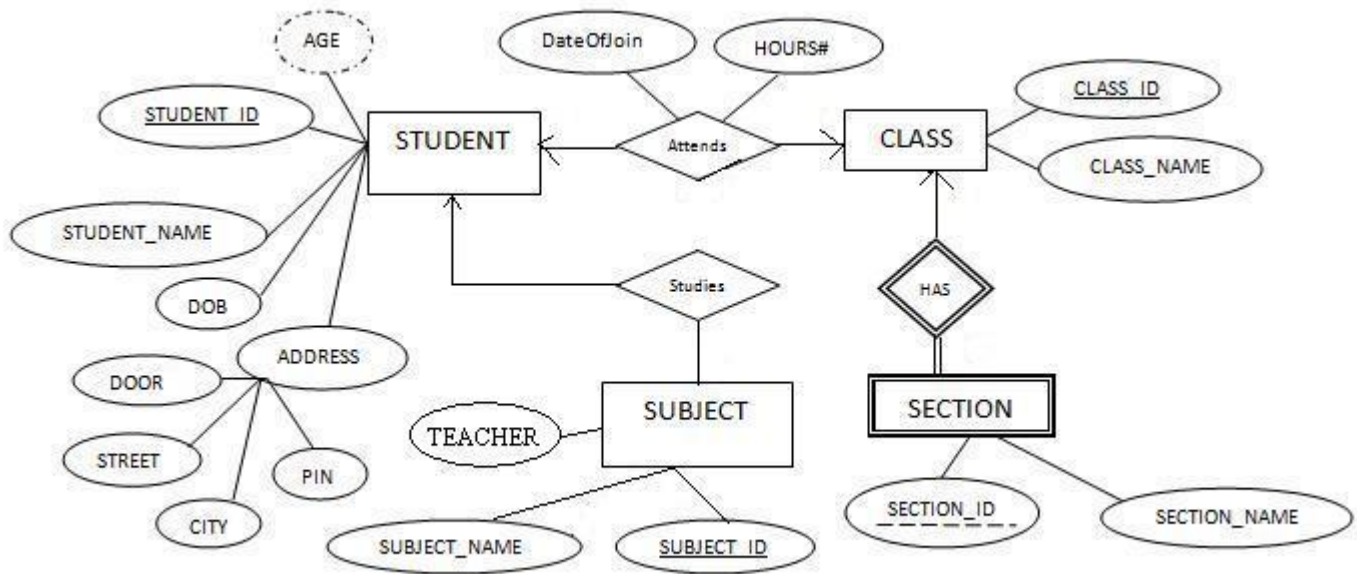
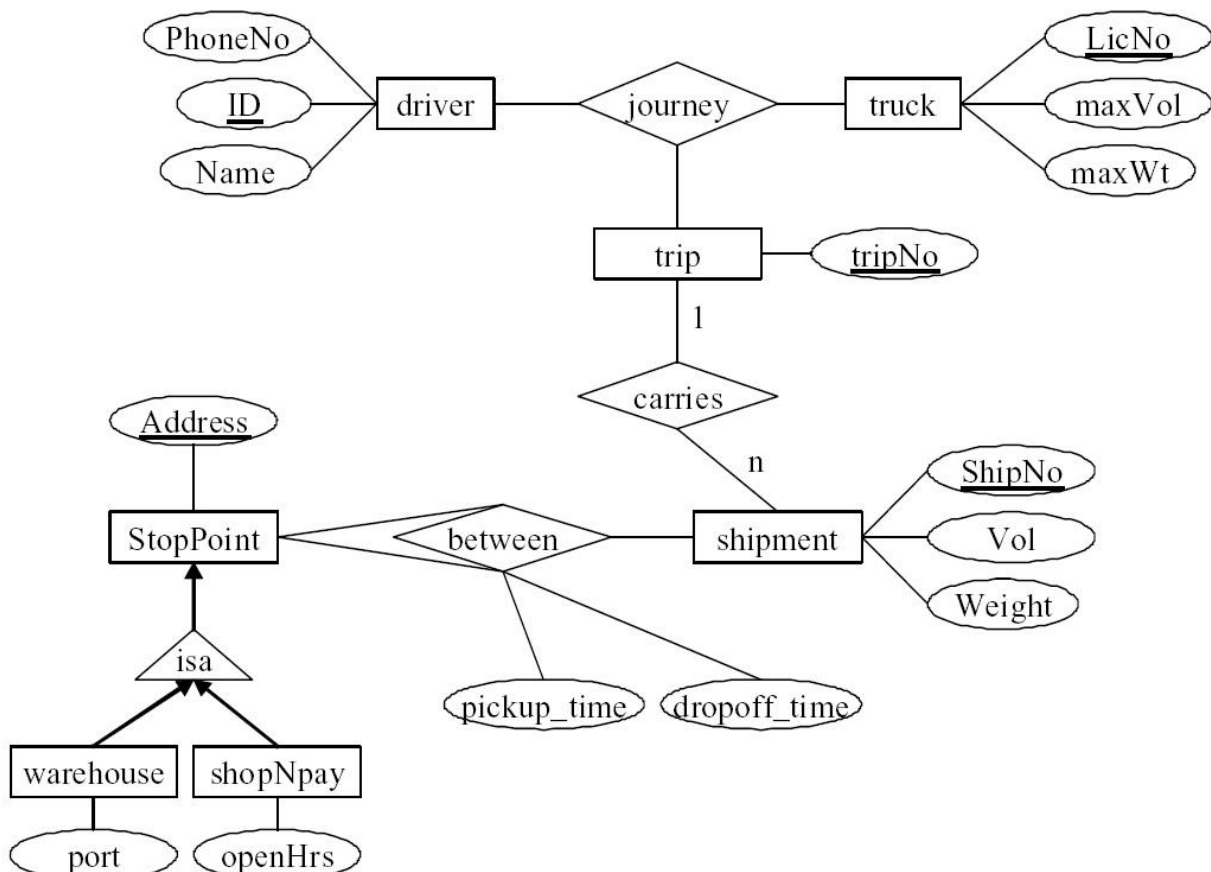


PART A**ER Diagram to Relational Mapping**

1. Convert the “Faculty of Technology” Entity Relationship (ER) Diagram you have drawn in Practical 05 into a Relational Schema Diagram.
2. Convert the following ER diagram into relation schema.



3. Convert following Entity Relationship diagram into Relational schema diagram.



4. Draw an ER diagram for the given scenario;

Suppose that you are designing a schema to record information about reality shows on TV. Your database needs to record the following information:

- _ For each reality show, its name, genre, basic_info and participants name. Any reality show has at least two or more participants.
- _ For each producer, the company name, company country. A show is produced by exactly one producer. And one producer produces exactly one show.
- _ For each television, its name, start year, head office. A television may broadcasts multiple shows. Each show is broadcasted by exactly one television.
- _ For each user, his/her username, password, and age. A user may rate multiple shows, and a show may be rated by multiple users. Each rating has a score of 0 to 10.

PART B

Use Employee table

1. Write a SQL statement to change the email column of employees table with 'not available' for all employees.
2. Write a SQL statement to change the email and commission_pct column of employees table with 'not available' and 0.10 for all employees
3. Write a SQL statement to change the email and commission_pct column of employees table with 'not available' and 0.10 for those employees whose department_id is 110.
4. Write a SQL statement to change the email column of employees table with 'not available' for those employees whose department_id is 80 and gets a commission is less than .20%
5. Write a SQL statement to change the email column of employees table with 'not available' for those employees who belongs to the 'Accounting' department.
6. Write a SQL statement to change salary of employee to 8000 whose ID is 105, if the existing salary is less than 5000.
7. Write a query to display the name (first_name, last_name) and department ID of all employees in departments 30 or 100 in ascending order.
8. Write a query to display the name (first_name, last_name) and hire date for all employees who were hired in 1987.
9. Write a query to display the jobs/designations available in the employees table in descending order.
10. Write a query to display the name (first_name, last_name), salary and PF (15% of salary) of all employees order by name in ascending order.
11. Write a query to display the last name of employees having 'e' as the third character and order the result by name in descending order.

Consider the customer table as sample data.

12. Write a query to get list of cust_city, sum of opening_amt, average of receive_amt and maximum payment_amt from customer table with following conditions-
 - grade of customer table must be 2,
 - average of receive_amt for each group of cust_city must be more than 500,
 - the output should be arranged in the ascending order of SUM(opening_amt),

13. Write query display cust_country and number of customers for the same grade for each cust_country, with the following condition -

- number of customer for a same 'grade' must be more than 2

Use salesman and customer tables.

14. From the given tables write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission.
15. From the given tables write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman, commission.
16. From the given tables write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission.
17. From the given tables write a SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer_id.
18. From the given tables write a SQL query to find those customers with a grade less than 300. Return cust_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer_id.
19. Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers. Use Right outer join.
20. From the given tables write a SQL query to find those customers with a grade less than 300. Return cust_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer_id. Use Left outer join.