DBMS LAB - I DDL

CASE STUDY - I

1. Create a table called "employees" with columns for "id", "name", "age", and "salary".
2. Add a column called "email" to the "employees" table.
3. Rename the "age" column in the "employees" table to "years_of_experience".
4. Delete the "salary" column from the "employees" table.
5. Create a table called "departments" with columns for "id" and "name".
6. Add a primary key constraint to the "id" column in the "employees" table.
7. Add a foreign key constraint to the "department_id" column in the "employees" table, referencing the "id" column in the "departments" table.
8. Add a unique constraint to the "email" column in the "employees" table.
9. Drop the "employees" table.
10. Truncate the "departments" table to delete all data, but keep the table structure intact.

CASE STUDY – II

1. Create a table called "customers" with columns for "id", "name", "email", and "phone_number".
2. Add a column called "address" to the "customers" table.
3. Rename the "phone_number" column in the "customers" table to "contact_number".
4. Delete the "email" column from the "customers" table.
5. Create a table called "orders" with columns for "id", "customer_id", "product_name", and "order_date".
6. Add a primary key constraint to the "id" column in the "customers" table.
7. Add a foreign key constraint to the "customer_id" column in the "orders" table, referencing the "id" column in the "customers" table.
8. Add a unique constraint to the "name" column in the "customers" table.
9. Drop the "orders" table.
10. Truncate the "customers" table to delete all data, but keep the table structure intact.
11. Create a table called "suppliers" with columns for "id", "name", "address", and "contact_number".
12. Add a column called "email" to the "suppliers" table.
13. Rename the "address" column in the "suppliers" table to "location".
14. Add a primary key constraint to the "id" column in the "suppliers" table.
15. Add a unique constraint to the "name" column in the "suppliers" table.

CASE STUDY - III

1. Create a table called "invoices" with columns for "id", "customer_id", "amount", and "due_date".
2. Add a column called "paid_date" to the "invoices" table.
3. Rename the "amount" column in the "invoices" table to "invoice_amount".

4. Delete the "due_date" column from the "invoices" table.
5. Create a table called "payments" with columns for "id", "invoice_id", "amount", and "payment_date".
6. Add a primary key constraint to the "id" column in the "invoices" table.
7. Add a foreign key constraint to the "customer_id" column in the "invoices" table, referencing the "id" column in the "customers" table.
8. Add a unique constraint to the "id" column in the "invoices" table.
9. Add a foreign key constraint to the "invoice_id" column in the "payments" table, referencing the "id" column in the "invoices" table.
10. Add a check constraint to the "amount" column in the "invoices" table to ensure that it is greater than zero.
11. Add a default value of "CURRENT_TIMESTAMP" to the "paid_date" column in the "invoices" table.
12. Drop the "payments" table.
13. Truncate the "invoices" table to delete all data, but keep the table structure intact.
14. Create an index on the "customer_id" column in the "invoices" table.
15. Add a NOT NULL constraint to the "customer_id" column in the "invoices" table.
16. Drop the primary key constraint from the "invoices" table.
17. Drop the foreign key constraint from the "customer_id" column in the "invoices" table.
18. Drop the unique constraint from the "id" column in the "invoices" table.
19. Drop the foreign key constraint from the "invoice_id" column in the "payments" table.
20. Drop the check constraint from the "amount" column in the "invoices" table.