Database Layer:

1. Points to be remember
2. Whenever we insert/delete/update table or table data then we commit the change.
3. String and date must be enclosed between single quotes (eg – ‘Ankita’ or ‘2000-12-16’).
4. Primary Key, Alternate Key, Candidate Key, Unique Key, Composite Key, Foreign Key.
5. Data Types and Operators.
6. Dealing with Date Type while selecting.
7. Joins.
8. Database Query
9. CREATE DATABASE [Database\_Name];
10. DROP DATABASE [Database\_Name];
11. BACKUP DATABASE [Database\_Name] TO DISK = ‘Disk\_Path’ [WITH DIFFERENTIAL];
12. ALTER DATABASE [old\_database\_name] MODIFY NAME = [new\_database\_name]; OR RENAME DATABASE [old\_database\_name] TO [new\_database\_name];
13. USE [Database\_Name];
14. Table Query
15. CREATE TABLE [Table\_name] (Col1 Datatype(size) constraints, Col2 Datatype(size) constraints, Col3 Datatype(size) constraints, Col4 Datatype(size) constraints, primary key (Col1), Foreign key (Col2) references table\_name (Col\_name));
16. DROP TABLE [Table\_name];
17. TRUNCATE TABLE [Table\_name];
18. ALTER TABLE [Table\_name] RENAME TO [new\_table\_name]; or RENAME [old\_table\_name] TO [new\_table\_name];
19. SELECT \* INTO [destination\_table\_name] FROM [source\_table\_name];
20. ALTER TABLE [table\_name] ADD (col1 Datatype(size) constraints, col2 Datatype(size) constraints..);
21. ALTER TABLE [table\_name] MODIFY [column\_name column\_type];
22. ALTER Table [table\_name] DROP COLUMN [column\_name];
23. ALTER TABLE [table\_name] RENAME COLUMN [old\_name] TO [new\_name];
24. Insert into [table\_name](col1,col2,….) values(val1,val2,….);
25. Delete from [table\_name] where [condition];
26. Update [Table\_name] set column\_name = new\_value where [condtion];
27. Data Query
28. Select \* from [table\_name];
29. Select \* from [table\_name] [WHERE condition] [GROUP BY column\_name] [ORDER\_BY column\_name ASC|DESC];
30. Select UNIQUE column\_name FROM [table\_name];
31. Select DISTINCT column\_name FROM [table\_name];
32. Select COUNT(expression/column\_name) FROM [table\_name];
33. Select AVG(number\_type\_column\_name) From [table\_name];
34. Select MIN(number\_type\_column\_name) From [table\_name];
35. Select MAX(number\_type\_column\_name) From [table\_name];
36. Select SUM(number\_type\_column\_name) From [table\_name];
37. Select TOP 2 \* From [Table\_name];
38. Select TOP 25 Percent \* From [Table\_name];
39. Select column\_name From [table\_name] Order by [column\_name] DESC Limit 1;
40. Select First(column\_name) From [table\_name];
41. Select Last(column\_name) From [table\_name];
42. Select column from table order by Rand() Limit 1;
43. Select [column\_name] AS [new\_column\_name] From [table\_name];
44. Select \* from [table\_name] where col1 in (Multiple Values);
45. Select table1.column1, table2.column1 from table1,table2 where table1.pk1 = table2.pk2;
46. Select \* from [table\_name] where conditon1 and condition2;
47. Select \* from [table\_name] where conditon1 or condition2;
48. Select \* from [table\_name] where colname between val1 and val2;
49. Select \* from [table\_name] where column like [a%, %a, %a%];
50. Select \* from [table\_name] where column is NULL;
51. Select \* from [table\_name] where column is NOT NULL;
52. SQL Joins
53. Inner Join – No Null
54. Left Join – Full Left Table
55. Right Join – Full Right Table
56. Full Outer Join – All Null
57. Cross Join – Both Table Values (No Null)

PL/SQL – Procedural Language / Structured Query Language

1. Functions

CREATE [OR REPLACE] FUNCTION function\_name

[(parameter1,parameter2,….)]

RETURN return\_datatype

IS

BEGIN

Executable\_section

[EXCEPTION

Exception\_section]

END [function\_name];

1. Procedures

CREATE [OR REPLACE] PROCEDURE procedure\_name

[(parameter1,parameter2,….)]

IS

[declaration\_section]

BEGIN

Executable\_section

[EXCEPTION

Exception\_section]

END [procedure\_name];

1. Triggers
2. Exception
3. Cursor
4. If, Case, Loops, Continue, GOTO
5. Variables and Constants