DBMS FACT SHEET

1) PRIMARY KEY CONSTRAINT ::

EG ::--> St_id int PRIMARY KEY

NOTE :: space between primary and key is required.

2) FOREIGN KEY CONSTRAINT ::

SYNTAX ::--> (1) CONSTRAINT FK_anything_anything
FOREIGN KEY (r.column) REFERENCES table.name (main.column) ,

(2) ALTER TABLE table.name
ADD CONSTRAINT FK_xyz_abc
FOREIGN KEY (r.column) REFERENCES table.name (main.column) ,

(3) ALTER TABLE table.name
DROP CONSTRAINT FK_xyz_abc ,

Note :: → Always use "constraint" because this is the only way to drop the fforeign key.
→ Space between foreign and key is required.

3) Check constraint ::

SYNTAX ::--> (1) CONSTRAINT CHK_anything_anything CHECK (____condition_____) ,

(2) ALTER TABLE table.name
ADD CONSTRAINT CHK_xyz_abc CHECK (____condition_____) ,

(3) ALTER TABLE table.name
DROP CHECK CHK_xyz_abc ,

Note :: → Always use constraint because this is useful for drop check.
→ we can use more than one column at one check constraint.
Eg; `CONSTRAINT CHK_Person CHECK (Age>=18 AND City='Sandnes') ,`

jetli kimmat check kervani hoy tetli vakht column name nakhvu .

eg :: constraint chk_gender1 check(Gender ='MALE' or Gender='FEMALE'),

---

4) Default constraint :: Jyare koi value insert keri na hoy to eni jagya e default value automatic set thay che……..

SYANTAX ::--> (1) **name d_type DEFAULT ‘_____’,**

Eg ::
```
City varchar(255) DEFAULT 'Sandnes',
OrderDate date DEFAULT GETDATE(),
```

(2) **ALTER TABLE** T_name
ADD CONSTRAINT DF_xyz
SET DEFAULT ‘_____’,

(3) **ALTER TABLE** T_name
DROP column_name
DEFAULT ‘_____’,

---

5) CREATE INDEX ::

SYNTAX :: (1) **CREATE INDEX I_name**
ON T_name (c1,c2,c3…..) ;

Eg ::
```
CREATE INDEX idx_lastname
ON Persons (LastName);
```

(2) ALTER TABLE T_name
DROP INDEX I_name;

---

6) CREATE VIEW ::

TABLE ni andar particular product banavvi hoy tyare……..
example tarke jo country na table ma ek evo view banavie k jo india no view khali nakhvama aave to ee country vishe badi info mali jay…..varmvar query nakhvama na aave.

SYNTAX :: (1) CREATE *view_name*
AS
SELECT *column1*, *column2*, …
FROM *table_name*
WHERE *condition* **;**

—>
```
CREATE [Brazil Customers]
AS
SELECT CustomerName, ContactName, City
FROM Customers
WHERE Country = 'Brazil ' ;
```

(2) DROP VIEW view_name**;**

(3) for show VIEW :: → SELECT * FROM view_name ;

7) SQL INJECTION ::

aama jyare user pase thi input mangvama aav tyare user input ni jagya e ek SQL STATEMENT enter karse je whole database ma run thase, programmer ni jan bahar.

(A) *SQL Injection Based on* ***1=1*** *is Always True* ***.***

Eg :: (

1) select * from student_detail where St_id=202101;

St_id	St_full_name	St_address	St_gender	Date_of_birth	St_landline_num	st_mobile_num	St_currant_sem	st_blood_group
202101	Pranav Anandkumar Rangoonwala	Surat	male	2004-04-05	1204562350	596541238	1	o

2) select * from student_detail where St_id=202101 or **1=1**;

St_id	St_full_name	St_address	St_gender	Date_of_birth	St_landline_num	st_mobile_num	St_currant_sem	st_blood_group
202101	Pranav Anandkumar Rangunwala	Surat	male	2004-04-05	1204562350	596541238	1	o
202102	Dev bimalbhai Lokhandwala	Navsari	male	2003-01-22	256478945	193574562	3	o
202103	Khushil nayanbhai Rana	Surat	male	2001-09-23	125463987	125479635	6	AB
202105	Riddhi Saileshbhai Patel	Jamnagar	female	1999-01-11	147896253	478523695	5	A
202106	Asmi Yashkumar Kapadia	Surat	female	2002-02-22	123698540	123647895	6	B+
202107	Monil Devanshbhai Jariwala	Jamnagar	male	2003-11-05	289631478	314789658	4	AB
202108	Kesar Maheshbhai kapadia	Navsari	female	2006-09-09	258412369	456891234	1	A-
202109	Sujal Ramanbhai Kinariwala	Surat	male	2004-12-31	126347895	784561230	2	B+
2021010	Umang pareshbahi Nayani	Surat	male	2003-05-17	1593574826	143692585	4	B-

(B) *SQL Injection Based on* ***OR “”=””*** *is Always True* ***.***

eg ::

(1) select * from student_detail where St_id=202101 **or ""=""** ;

St_id	St_full_name	St_address	St_gender	Date_of_birth	St_landline_num	st_mobile_num	St_currant_sem	st_blood_group
202101	Pranav Anandkumar Rangunwala	Surat	male	2004-04-05	1204562350	596541238	1	o
202102	Dev bimalbhai Lokhandwala	Navsari	male	2003-01-22	256478945	193574562	3	o
202103	Khushil nayanbhai Rana	Surat	male	2001-09-23	125463987	125479635	6	AB
202105	Riddhi Saileshbhai Patel	Jamnagar	female	1999-01-11	147896253	478523695	5	A
202106	Asmi Yashkumar Kapadia	Surat	female	2002-02-22	123698540	123647895	6	B+
202107	Monil Devanshbhai Jariwala	Jamnagar	male	2003-11-05	289631478	314789658	4	AB
202108	Kesar Maheshbhai kapadia	Navsari	female	2006-09-09	258412369	456891234	1	A-
202109	Sujal Ramanbhai Kinariwala	Surat	male	2004-12-31	126347895	784561230	2	B+
2021010	Umang pareshbahi Nayani	Surat	male	2003-05-17	1593574826	143692585	4	B-

8) SQL PARAMETERS ::

→ to protect a web site from SQL injection, you can use SQL parameters……..
→ parameter are represented in sql using @ .

9) SQL HOSTING ::

→ aapdi website ma user na data save kerva and retrieve kerva ek server ni jaroor pade.
→ if your web server is hosted by an Internet Service Provider (ISP), you will have to look for SQL hosting plans.

QUERY SYNTAX ::

(1) SELECT ::

—> **SELECT** _____ ,____
FROM *table_name*;

—> **SELECT * FROM** *table_name*;

(2) DISTINCT :: DUPLICATE VALUE **ELIMINATE** KARE.

—> **SELECT DISTINCT** _____ , _____
FROM *table_name*;

(3) WHERE ::

→ **SELECT** ____, ____
FROM *table_name*;
WHERE *condition* ;

→ **SELECT** ____, ____
FROM *table_name*;

WHERE NOT *condition ;*

→ **SELECT** ____, ____
FROM ***table_name*;**
WHERE *condition* ***AND*** *condition* ***AND*** *condition ;*

→ **SELECT** ____, ____
FROM *table_name*;
WHERE *condition* **OR** *condition* **OR** *condition ;*

→ **SELECT** ____, ____
FROM *table_name*;
WHERE NOT *condition AND NOT condition ;*

(4) INSERT INTO ::

→ **INSERT INTO** *table_name (column je select karvi hoy)*
VALUES *(value1 , value 2, value 3) ;*

(5) UPDATE ::

→ **UPDATE** *table_name*
SET ***C1=" ", C2=" "***
WHERE *condition ;*

(6) DELETE ::

→ **DELETE**
FROM table_name
WHERE condition **;**

(7) ORDER BY ::

(A)→ SELECT TOP 3
FROM *table_name*
WHERE *condition* **;**

```
—> SELECT TOP 3 * FROM Customers
   WHERE Country='Germany';
```

(B)→ SELECT column_name
FROM table_name
ORDER BY column_name **ASC ;**

→ **SELECT** *
FROM table_name

ORDER BY column_name **ASC ;**

```
—> SELECT * FROM musicians
   ORDER BY age DESC, instrument ASC;
```

(8) FETCH & LIMIT ::

→ **Both have same functionality.**

(A) SELECT *
FROM table_name
LIMIT 3 ;

(B) SELECT *
FROM *table_name*
FETCH *first 3 rows only ;*

(9) MIN, MAX, COUNT, AVG, SUM ::

→ **all have same syntax.**

(A) SELECT MAX( )
FROM *table_name ;*

(B) SELECT MAX ( )
AS *XYZ*
FROM *table_name* **;**

(10) LIKE OPERATOR ::

→ **SELECT ***
FROM *table_name*
WHERE *condition*
LIKE ‘ uma%‘ ;

(11) IN OPERATOR ::

→ **WHERE CLAUSE ni andar MULTIPLE VALUE add kerva mate.**

(1) → SELECT *
FROM *table_name*
WHERE *condition*
IN (value1 , value2, value3 , …) **;**

eg ::
```
SELECT * FROM Customers
WHERE Country IN ('Germany', 'France', 'UK');
```

→
```
SELECT * FROM Customers
WHERE Country
IN (select ___ from ___ );
```

(2) → **SELECT** *
FROM *table_name*
WHERE *condition*
NOT IN (value1 , value2, value3 , …) ;

eg ::
```
SELECT * FROM Customers
WHERE Country NOT IN ('Germany', 'France', 'UK');
```

(12) BETWEEN ::

→ **SELECT** *
FROM table_name
WHERE column_name
BETWEEN VALUE1 **AND** VALUE2 ;

eg ::
```
SELECT *
FROM Products
WHERE Price
BETWEEN 10 AND 20;

SELECT *
FROM Products
WHERE Price
BETWEEN 10 AND 20 AND CategoryID NOT IN (1,2,3);

SELECT *
FROM Products
WHERE ProductName
BETWEEN 'Carnarvon Tigers' AND 'Mozzarella di Giovanni'
ORDER BY ProductName;
```

(14) ORDER BY ::

→ Jyare same value ni row ne bhegi karvi hoy tyare GROUP BY vapray.

→ **SELECT** *Column_name*
FROM *Tablel_name*
WHERE *Condition*
GROUP BY *Column_name*
ORDER BY *Column_name ;*

Eg ::------> **SELECT COUNT**(CustomerID), Country

FROM Customers

GROUP BY Country

ORDER BY COUNT(CustomerID) **DESC**;

(13) JOIN ::

(A) INNER JOIN ::

Join matching record of two tables.

SYNTAX ::

—----->**SELECT** *table_name1.column_name1 , table_name2.column_name2*
FROM *table_1*
INNER JOIN *table_2*
ON *table_1.same_column = table_2.same_column ;*

Eg ::

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers
ON Orders.CustomerID = Customers.CustomerID;
```

—----->**SELECT** *Column_name*
FROM *table_1*
INNER JOIN *table_2*
ON *table_1.same_column(1) = table_2.same_column(1)*
INNER JOIN *table_3*
ON *table_1.same_column(2) = table_3.same_column(2) ;*

Eg ::

```
SELECT Orders.OrderID , Customers.CustomerName , Shippers.ShipperName
FROM Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID
INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID ;
```

(B) OUTER JOIN ::

(1) LEFT OUTER JOIN
(2) RIGHT OUTER JOIN
(3) FULL OUTER JOIN

SYNTAX ::

—> **SELECT** *table_name1.column_name1 , table_name2.column_name2*
FROM *table_1*
LEFT OUTER JOIN *table_2*
ON *table_1.same_column = table_2.same_column ;*

—> **SELECT** *table_name1.column_name1 , table_name2.column_name2*
FROM *table_1*
RIGHT OUTER JOIN *table_2*
ON *table_1.same_column = table_2.same_column ;*

—> **SELECT** *table_name1.column_name1 , table_name2.column_name2*
FROM *table_1*
FULL OUTER JOIN *table_2*
ON *table_1.same_column = table_2.same_column ;*

(14) UNION ::

SYNTAX ::

SELECT column_name **FROM** table_name
UNION ALL
SELECT column_name **FROM** table_name **;**

—--> Eg ::

```
1) SELECT City FROM Customers
   UNION ALL
   SELECT City FROM Suppliers
   ORDER BY City;

2) SELECT City FROM Customers WHERE condition
   UNION ALL
   SELECT City FROM Suppliers WHERE condition
   ORDER BY City;
```

(15) HAVING ::

WHERE clause ee **AGGREGATE** sathe vapratu nathi tethi where ni jagya e **HAVING** function vapray che.

SYNTAX ::

SELECT *column_name(s)*
FROM *table_name*
WHERE *condition*
GROUP BY *column_name(s)*
HAVING *condition*
ORDER BY *column_name(s)* ***;***

—--> Eg ::

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5;
```

(15) ANY & ALL :: ANY subquery ma useful che……

` subquery check kerva , true & false value check kerva.

OPERATOR —> (=, <>, !=, >, >=, <, or <=)

SELECT *column_name(s)*
FROM *table_name*
WHERE *column_name*
OPERATOR
ANY(SELECT *column_name* FROM *table_name* WHERE *condition*) **;**

Eg ::

```
SELECT ProductName
FROM Products
WHERE ProductID = ANY
 (SELECT ProductID
  FROM OrderDetails
  WHERE Quantity = 10);
```

(15) DECIMAL / DEC / FIXED ::

column_name DECIMAL (T,D)

T = TOTAL DIGITS,
1234.55 —> T= 6 ; RANGE —> 0-65

D = DIGIT AFTER DECIMAL
1234.55 —> D= 2; RANGE —--> 0-30

EG → 1234.55 —--> DECIMAL(6,2)

16) Sub query :::

—>without aggregate function ::

```
SELECT * FROM T.name
  WHERE same.column IN (SELECT same.column FROM T.name
    WHERE income > 350000);
```

–----> WITH aggregate function ::

```
SELECT <column which u want with aggregate function >
FROM T.name
 WHERE same.column = (SELECT MAX(same.column) FROM
employees);
```

NOTES ::::

1> I can use anything for declare character in query ‘__’ or “__” .

2> Select mySQL based only on month and year .

—> SELECT * FROM projects
WHERE **YEAR(Date)** = 2011 AND **MONTH(Date) = 5** ;

3>
—> SELECT ENAME FROM EMP WHERE LENGTH(ENAME) = 5;

4> EK TABLE NA DATA BIJA TABEL SATHE CONNECT KERVA …

—>

```
SELECT PRODUCT_MST.PROD_NAME, PROD_DESC, PROD_RATE , SALES_ORDER_DTL.SALES_DATE, CUSTOMER_NAME,ORDER_CITY
 FROM PRODUCT_MST
JOIN SALES_ORDER_DTL
ON PRODUCT_MST.PROD_ID=SALES_ORDER_DTL.PROD_ID
WHERE SALES_ORDER_DTL.SALES_ID > 2;
```

5> find 2nd / 3rd / 4th most max / min ????

—-->
tyrre pella je 1st lowest hoy te sodhvanu tyar bad tenathi moti value hpy teni condition mikvani.

2nd lowest ::

```
select * from EMPLOYEE
where salary = (select min(salary) from EMPLOYEE
where salary > (select min(salary) from EMPLOYEE) ;
```

3rd lowest ::

```
select * from EMPLOYEE
where salary = ( select min(salary) from EMPLOYEE
where salary > (select min(salary) from EMPLOYEE
```

```
where salary > (select(min(salary) from EMPLOYEE)) ) ;
```

6> RENAME TABLE ??

ALTER TABLE _____
RENAME TO __new_name____ ;

7> drop column ???

ALTER TABLE _____
DROP COLUMN _____ ;

8> RENAME COLUMN NAME ??

ALTER TABLE _____
COLUMN RENAME _____ TO _____ ;

ALTER TABLE _____
CHANGE ___old_c_name____ ___new_c_name___ <datatype> ;

9> add column ???

ALTER TABLE_____
add column _____ ;

****** note ******

foreign key banavva mate pela tene primary key declare kerva ;

foreign key vada data ne pela tena main table ma value insert karvi .tyar bad tena reference table value insert karvi .

Uni

****** Error *******