

SQL Basics → DPL, DML, DCL, TPL
 SQL (Structured Query Language) 1970s
 ↳ High Level Language
 ↳ Platform independent and portable
 ↳ Standardized and fast.

Date: / /

CREATE
ALTER
DROP
Truncate

DDL

DML

INSERT

SELECT

UPDATE

DELETE

REVOKE
GRANT

DCL

TCL

ROLLBACK

COMMIT

CHECKPOINTS

Ques:

Q1. DCL TCL

Q2.

Q3.

Q5. To save the changes permanently in the database.

1

TCL → COMMIT is used.

Q6.

V.V.Imp

↓

[SQL DataTypes]

#	Input	Used Data Type	Stored As	Reason
1.	134	INTEGER	134	int ← decimal
2.	234.6	INTEGER	234	Truncates after decimal
3.	134	NUMERIC	134	Represented as int is -
4.	234.6	NUMERIC	135	Round off towards nearest neighbor.
5.	134	NUMERIC(2)	Data Exception	Results in data loss so, it gives exception.
6.	23.6	NUMERIC(2)	24	Round off towards nearest neighbor.
7.	23.5	NUMERIC(2)	23	if equidistant then round towards ③ Nearest neighbor.
8.	23.4	NUMERIC(2)	23	RAJDHANI

Just
Rulls

Date: / /

9. 99.84 NUMERIC(4,1) 99.8 Round towards nearest neighbor

10. 99.85 NUMERIC(4,1) 99.8 Round towards zero

11. 99.86 NUMERIC(4,1) 99.9 Round towards nearest neighbor
~~100~~ * no consider tenth

12. 99.98 NUMERIC(4,1) 100.0 Round towards nearest neighbor.

13. 999.96 NUMERIC(4,1)
~~1000.0~~ As rounding off to 1 decimal digit

Data Exception

results in

1000.0

and maximum precision is 4 and 1 is mandatory as scale
→ so it is not possible

Data types in SQL : (Types)

Character
Numeric
Miscellaneous.

Character



CHAR

- Can be used without specifying the length and default length is 1

To store,

names, city, class, etc.

VARCHAR

- Should always be used with length (X error otherwise)

default +

* mandatory to give

Date: / /

CHAR (L)

USE

Used for storing fixed length strings

VARCHAR (L)

Used for storing variable length strings

Storage characteristics

Fixed L characters are allocated for storage. Trailing spaces are added if data to be stored is less than L.

The space used for each stored item is proportional to its actual length.

Example

NAME CHAR(10) field will

store "Clark" as 10 bytes by appending 5 trailing spaces.

NAME VARCHAR(10) will

store "Clark" as 5 bytes only.

Alternate
Name

CHARACTER (L)

CHARACTER VARYING (L)

LONG VARCHAR (L)

Numeric data types

TINYINT, SMALLINT, INTEGER, BIGINT, NUMERIC, DECIMAL, DOUBLE, REAL, FLOAT.

Not mandatory to give:

i.e. used only NUMERIC if necessary
like normal INTEGER

But

INTEGER

(for storing whole nos.)

NUMERIC (P, S)

(m stores decimal value)

Maximally
8 bit
precision

32-bits

(2⁸, 32) bits

Default:

32-bits

(2⁸, 0) bits.

T.

that is why 100.0

* Range

-2,147,483,648 — 2,147,483,647

↓
is stored as
100

NUMERIC() ≡ DECIMAL()

Mantissa \rightarrow scale dekho pehle \rightarrow fir bas jeh dekhta
 Waise hisaab se decimal
 Se petile kitne aasute hain Date: / /

- * Precision and Scale of non-Integral datatype determine the maximum value that can be stored in it.

Precision
 +
 NUMERIC (3, 1) Scale

* NUMERIC (3)
 default
 Scale is 0

is NUMERIC (3, 2)
 0

Precision
 1 2 3
 [9] . [9] [9]

2 Scale

Precision
 NUMERIC (3, 1)
 1 2 3
 [9] [9] . [9]
 1 Scale

Precision
 NUMERIC (9, 3)
 1 2 3
 [0] . [9] [9] [9]
 1 2 3 Scale

* Wrap

- * If No Scale is given, then NUMERIC datatype is used to store ~~NUMBER~~ INTEGRAL values

NUMERIC(4, 1) 100.0 \rightarrow is stored as 100

Other Data Types:

V.V.V.V.V IMP

Date: / /

* DATE YYY-MM-DD format
like '2008-08-22'

* TIME HH-MM-SS format like
'20:08:08'

* TIMESTAMP Combination of Date and Time Date of Birth
'2008-08-08' 20:08:08'

* CLOB (Character Large object) Text can't fit in
CHAR/VARCHAR

Length limit can be given in units of Kilobyte

Default size is 1 GB

Example

Resume CLOB
(30K)

* BLOB (Binary Large Object)

for storing Large binary data
like Movies, images etc.

Some:

* BOOLEAN for storing TRUE or FALSE

* Unknown values using NULL or "UNKNOWN"
"UNKNOWN" in
Single Quoted

Type Conversion

Date: / /

* 1 Implicit conversion:

- ↳ In an expression having different but compatible data types,
- ↳ An SQL type "wide" enough to store the result is returned.

Implicit conversion is used in Assignment, Comparison and Aggregation.

Example : Any combination of NUMERIC and INTEGER returns NUMERIC/DECIMAL.

* 2 Explicit conversion CAST is used.

Syntax : CAST (<value> as <datatype>)

Example CAST ('MANGO' as CHAR(3))

↓

'MAN'

* Imp.

When the result or expression is casted to some type ~~but~~ that, an exception is thrown if the resulting value cannot be stored in the column

Same for CHAR(6) as well

Tested ✓

Date: / /

Took

Data type used

Stored as

Records

'Hope.'

VARCHAR(6)

'Hope.'

Input contain 4 letters

+ 3 spaces but only

6 characters are allowed

So 1 many space is

truncated.

'Grammati'

VARCHAR(6)

'Gramma'

data exception

✓ T.M.P

Truncation takes place
only for extra trailing
spaces

Input length exceeds size limit

No truncation takes place.

'Run'

CHAR(6)

'Run - - -'

CAST

('Grammati' as)
CHAR(4)

CHAR(6)

~~'Grammati'~~

'GIRAM - - -'

CAST

('Grammati' as)
CHAR(4)

VARCHAR(6)

'GIRAM'

CAST

('Smile' as)
CHAR(3)

VARCHAR(6)

'smi'

Cast ('we' as CHAR(4))

VARCHAR(6)

'we - - -'

CAST ('Wind' as
VARCHAR(6))

CHAR(3)

Wind

→ data exception
size limit: 3
provided 4,

RAJDHANI

* $\text{CAST}(\text{'Wind'} \text{ AS} \text{ VARCHAR}(2))$ \rightarrow CHAR(3) \rightarrow 'WII'

Quiz - Sql data types!

119. 333

Q1. ~~DEC(3,2)~~ (6,3)Q2. NUMERIC(4,2) \rightarrow Max value

99.99

Q3: Scale \rightarrow no. of digits allowed after decimal point
 Precision \rightarrow total no. of digits allowed before and after the decimal point.

Q4: Book Title \rightarrow VARCHAR(50)

(Don't take examples literally)

- * Points for Numeric data types (INTEGER) (NUMERIC)
 - Decimal values to INTEGER → Decimal values Truncated.
 - Decimal values to NUMERIC → Round-1104-Below Date: 1 1
 (where scale is 0)

"Round towards "Nearest Neighbor""
 if both neighbors equidistant → round towards zero,
 (i.e. lower side).

- 3) If the Non-Fractional part of the value cannot be represented in the new type, cast off conversion cannot take place → data exception.

By default → scale = 0

Precision - 128 bits.

Important

- * INTEGER → no length to be specified (-)
 ↗ even mon.

- * If we want fixed size integer then we can use NUMERIC(3) or NUMERIC(7)
 etc



Quiz problem 2nd last + → Quiz - SQL Data Types

* Operations and Expression :

$$\downarrow \\ 3 + 4 * 5 * (9 - 7) = 43 \\ 3 + 4 * 5 * (2)$$

Date: / /

$$3 + 20 * 2$$

$$3 + 40$$

$$\underline{43}$$

* These Expressions evaluate to a Single value and are used in SELECT and WHERE clauses.

* ARITHMETIC OPERATORS:

*	$15 + 5$	20
-	$15 - 5$	10
*	$15 * 5$	75
/	$15 / 5$	3

* Points while working with Arithmetic operators:



1) When a Decimal value is added or subtracted with another decimal or integral value, the resulting scale is the Higher Scale values of all the terms.

$$\begin{array}{r} 34.587 \\ + 7.850 \\ \hline 42.437 \end{array}$$

$$7.80 + 34 = \underline{41.80}$$

$$\begin{array}{r} 34.58 \\ - 7.8594 \\ \hline 26.7206 \end{array}$$

$$\begin{array}{r} 34.5800 \\ - 7.8594 \\ \hline 26.7206 \end{array}$$

2) When a decimal value multiplied by Decimal or Integral value, the resulting scale is the sum of the scales of the two terms.

$$7.80 * 34.587 = 269.77860$$

$$34.58 * 7.8500 = 271.453600$$

$$34.58 * 7.85 = 271.4530$$

$$3 \overline{)7}^{2:3} \\ 3 \overline{)7}^{2:3}$$

Date: / /

scale

- By default the result of division and Avg aggregate function is same as higher scale value of the terms involved.

$$7/3 = 2 *$$

$$7.0/3 = 2.3 *$$

$$7.0/3.00 = 2.33 *$$

↓

But here to avoid confusions

↓

SQl AVG SCALF is set to 5:

- When two values will be divided. The resulting scale will be the maximum of:
 - > highest scale value of terms
 - or
 - > decimal scale of 5.

2★ Relational operators:

Equal check =

Not Equal to <> <> *

>= >
<= <

Assignment → =

Equality check → =

***** An Arithmetic Operation involving NULL always return NULL *****

- * 3) By default the result of division and Avg aggregate function is same as higher scale value of the terms involved.

$$7/5 = 2 *$$

$$7.0/3 = 2.3 *$$

$$7.0/3.00 = 2.33 *$$

↑

But here to avoid confusions

↑

SQl Avg Scale is set to 5.

∴ When two values will be divided. The result's scale will be the maximum of:

1) highest scale value of terms
or

2) decimal scale of 5.

Relational operators:

Equal check =

Not Equal to <> ← *

>=

7 = >

<= <

Assignment → =

Equality check → =

***** An Arithmetic Operation involving NULL always return NULL *

5 + NULL → NULL
6 * NULL → null

0 / NULL =

Date: / /

*3 LOGICAL OPERATORS :

1) AND → true if both conditions true

2) OR → true if any of them true

3) NOT → negate the true to false
false to true

Department NOT IN

('HR', 'Finance')

*4 Special Operators:

both inclusive

⇒ BETWEEN <lower limits> AND <upper limits>

2) IN (List of values) → Matches if a value lies in the given list of values.

3) LIKE → Matches a character pattern. Suppose like

'S%o'

4) IS NULL Checks if the value is NULL Bonus is NVL

* Practice - Quiz - SQL Basics

Q1: NUMERIC(4,1)

(Truncation and all Round off)
dhejoon ch Rakhi hai

1) 99.5 ✕

★ Data Exception occurs only if Non-Fractional part is greater than the storage size.

2) 999.98 ✕

1000.0 ✕

3) 9999.9 ✕

10000.0 ✕

4) 98.47

98.5 ✕

5) 9.7856

9.8

Check Zada scene nahi hai simple

RAJDHANI

Q2 CLOB (If then,
run simply go for CLOB)

Q3 INTEGER (Jyada nahi sare hoga
about fixed size and all here
age aur he aayegi)
Date: / /

Q4 * fixed size INTEGER value chahiye toh
~~NUMERIC(6)~~

Q5 DEPT NOT IN ('Design', 'Sales', 'Sale') Dept=sales
NOT true
(Ans) false

Q6 NOT (S1=5) OR (DEPT NOT IN ('DEP', 'XYZ')
AND SALARY BETWEEN 30000
AND 50000)
true or (True AND true)

* \leftrightarrow or $!=$
done chalte
done

Q7 * Important

DDL commands is used to modify the database objects in SQL?

★ DML

1. INSERT Inserting Values into Table

Go to the backside of the copy

and see DBMS Insert points

V.V-Fmp

2★ UPDATE↳ **Update** table-name **Set** salary = salary * 1.1↳ Multiple columns update using comma,↳ where condition to update multiple rows

★ Death is clear
 COPY me DBMS part
 me don't have maine
 chize with d → joker
 Doubts and all
 be care please -

★ Quiz - Update Data

Q3

Alan F 2018-02-01

SM

Fmp.

★ DDL

★ CREATE TABLE

Database objects → Table columns

(Not data
of Rows)

Date:

/

★ Create Table Syntax:

DDL (Data Definition Language) commands are used to create, modify and drop database objects.

* Create table

- 1) Each table → must have a Name → only no. of columns (At least 1) minimum
- 2) Each Column → must have a datatype.
- 3) CREATE TABLE fails if table with that name already exists
- 4) All tables → Unique names
- 5) Database cannot have duplicate name tables.
- 6) No two columns also have the same Name

CREATE TABLE Student (
StudentId INTEGER,

DOJ DATE.
);

→ Column Names Datatype, another col

* Create table Syntax Errors :

3 errors to be found -

CONSTRAINTS: "rules that regulate the data entering into tables and how they are related!"

↳ for robust and reliable databases.

* CONSTRAINTS are added while creating the Table.
OR AFTER Table

• NOT NULL: Data Integrity

• PRIMARY KEY: Data Integrity

• CHECK: Data Integrity + Business Rule.

• UNIQUE: ↓ prevents duplicate entries → Data Integrity.

Allows Multiple NULL VALUES

• FOREIGN KEY — CAN BE DUPLICATES

↓ CAN BE NULL

Relationship b/w tables → Referential Data Intgrity

Advantages

1) Restrict

2) Enforce

3) Prevent invalid or inconsistent data.

* Based on no. of columns they act on + place where they are defined

↳ Column-Level: Specific to an individual column
Non-inter-column relationship.

a) Table-Level if constraint involves multiple columns

↳

must be defined at Table Level

i.e. anywhere in the table after the required columns have already been defined.

Suggested place → after all the columns are defined.

* Imp → 1) Single column constraint → Column Level / Table Level

→ 2) Composite Constraint → Table Level only.
(involving more than one column)

→ 3) NOT NULL → only at COLUMN LEVEL RAJDHANI

Constraint Type	Applies on
Single Column Constraint	Single column
Composite Constraint	Multiple Columns
Date: / /	

Constraint Type	Where
Column Level constraint	With column definition / the table
Table Level constraint	Other column definition.

Constraints Examples:

↓
CREATE TABLE Student (

StudentId INTEGER CONSTRAINT Student-const_pk

StudentName VARCHAR(25) NOT NULL,

DOS DATE .

);

* Constraint Name is optional and can be given using CONSTRAINT keyword

* If the CONSTRAINT keyword is used, then name must be given,

* Constraint name is ~~can~~ must be unique hence advised to follow the convention :

↓

tablename - const - pk

★

It is always advised as best practice to give constraint names always for creating a well-structured and maintainable database.

especially for Foreign Key, UNIQUE KEY, and PRIMARY KEY constraints.

NOT NULL

(Because you might have to drop or alter them)

★

More than 1 constraint can also be defined for a single attribute (spaces in b/w).

All column level constraints except NOT NULL can be defined in the Table Level but no table level constraint can be given as column level constraint.

Date: / /

- 1) NOT NULL
- 2) PRIMARY KEY
- 3) CHECK
- 4) UNIQUE
- 5) FOREIGN KEY
- 6) DEFAULT.

PRIMARY KEY

1) NOT NULL \rightarrow prevent the column from accepting NULL values

This constraint can only be applied as a column level constraint.

Example:

> CREATE TABLE Student (

StudentId INTEGER CONSTRAINT stu_id
NOT NULL,

FName VARCHAR(10) NOT NULL,

LNAME VARCHAR(10)

);

> INSERT INTO Student VALUES (1001,

'Alex', 'Ferguson');

1 row(s) affected.

> INSERT INTO Student VALUES (1001, NULL,
Ferguson);

X Integrity Constraint Violation: NOT NULL check en-

spale in NOT NULL

> INSERT INTO Student VALUES (1002, ' ', 'Fu')

1 row(s) affected

empty string in NOT NULL

> INSERT INTO Student VALUES (1003, "", 'Fu')

1 row(s) affected

*2 PRIMARY KEY

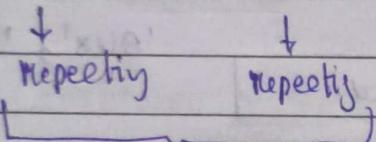
- * NOT NULL UNIQUE
- * Only 1 PRIMARY key in a Table

- > Insert valid ✓
- > Insert valid ✗
- X > Insert a NULL in PK ✗
- X > Insert duplicate in PK ✗

COMPOSITE PRIMARY KEY

- ↳ is a PRIMARY KEY only with multiple attributes.

StudentId, CourseId



- * Some rule also

*3 CHECK CONSTRAINT

→ To limit the values that can be specified for a column.

Date: / /

CREATE TABLE Student (

```
StudentId INTEGER,  
FName VARCHAR(10),  
Gender CHAR(1) CHECK (Gender IN ('M', 'F'))
```

);

- ↳ > Insert a record with permitted value.

> `INSERT INTO Student VALUES (1001, 'IFlex', 'M');`

✓

- > Any other value than 'M' or 'F'
- X Integrity constraint violation, check

caused by

IMP

NULL is allowed.

CREATE TABLE RESULT (

```
StudentId INTEGER,  
CourseId INTEGER,  
Marks NUMERIC(3))
```

CONSTRAINT result-PK PRIMARY KEY (StudentId, CourseId)

);

IMP

★ NO NULL values for any attribute in the primary key.

X integrity constraint violation:
unique constraint.

UNIQUE constraint
ensures that two rows in a ~~other~~ table cannot have same value in that column.

unlike PRIMARY KEY.

• Allow NULL (multiple NULLS inserted in the CHILD tables.)

• A table can have many UNIQUE constraints

NULL is Specially Treated in UNIQUE

Exercise PDL-Create-Product

#

* Need for FOREIGN KEY

we don't want Date: / /

incorrect/misleading data to be

inserted in the CHILD tables.

Data about the student rows,
a course may be added to

the Course table even though
that student is not there in
the Student table.

CREATE TABLE Result (

StudentId) INTEGER CONSTRAINT

result_id_pk REFERENCES

student (studentId)

Table Level

CONSTRAINT result_stu_pk

FOREIGN KEY (studentId)

REFERENCES student (stuId)

)

• Imp NULL values allowed simply.

Exercise #

Exercise #

★ DEFAULT

So far we have seen the constraints that restrict the values of columns to maintain

Data Integrity

* Sometimes [Data Integrity] can be achieved indirectly by suggesting valid values.

→ While insertion using column names, if no value is given for such a column then that default value will be taken

IMP: This constraint (not actually) do not prevent inserting any value

IMP

- Data type of the column and Default expression must be same.

- can be provided for Nullables as well as NOT NULL attributes

- 1) User supplies a value that value is taken



2) Inserted NULL

→ NULL will be taken

- 3) No value specified
Default value taken

→ CREATE TABLE student

StudentId INT,

Date DATE DEFAULT SYSDATE

);

common errors :



Important
Game

(to do it)

* Tryout. Create table

↓
find Syntax Errors

↓
1) VARCHAR (L)

Mandatory

2) REFERENCES

Nhi likha hua tha



Dhyaoon se derhna hai]

CREATE TABLE and Constraints - Quiz
An attribute allows NULL values unless NOT NULL constraint is used

3 wrongs go read

Date: / /

Important questions all go and do.

* ALTER TABLE is the DDL command (in two clauses)

DDL

ALTER TABLE

(Add/ Remove / Modify)

Edtta type Change constraint
one or more constraints.

Columns -

(CLUSES TO BE USED)

* Adding a Column : (ADD COLUMN clause) or ADD

> ALTER TABLE Employee ADD COLUMN
Address CHAR(30);

* 5 Adding a Constraint

on existing table.

(ADD CONSTRAINT clause)

> ALTER TABLE Employee ADD
CONSTRAINT emp-id-PK

PRIMARY KEY (employeeId);

UNIQUE name for a constraint

* 2 Modifying Column (Datatype/size)
(ALTER COLUMN clause)

> ALTER TABLE Employee ALTER
COLUMN Address VARCHAR(30);
Similar to increase or decrease

* 6 Deleting a Constraint

(DROP CONSTRAINT clause)



> ALTER TABLE Employee

DROP CONSTRAINT emp-id-PK;

> ALTER TABLE Employee ALTER
COLUMN Address VARCHAR(10);

* 3 Renaming a Column (RENAME TO
clause is used)

> ALTER TABLE Employee ALTER COLUMN
Address RENAME TO NewAddress;

Imp Points

* while changing the datatype
or size of the column
the existing data should
be compatible
otherwise X
error.

along with

ALTER COLUMN
clause.

* If the
column does
not contain
any data
then no
error.

* 4 Deleting a Column

Along with (DROP COLUMN)

(DROP COLUMN clause)

OR (DROP

> ALTER TABLE Employee ALTER COLUMN

DROP COLUMN address;

1) Dropping column → Cannot be recovered back
JMP ~~**~~ → There must be at least one column present in the table after dropping the columns.

Date: / /

DROP COLUMN = DROP only if you try

X failed: need 3 column or cannot delete sole column of the table.

JMP

* If you Add a Constraint to a Column and that column already contains values that violates that constraint then it will throw the error.

Example: if StudentId contains 1001 and you add a constraint

X We can DROP PRIMARY KEY →

X Integrity constraint violation

② Adding column:

- * ① The new column becomes the Last column
- ② The existing rows will contain null value for the newly added column.

③ Modifying the Column

- Datatype can be changed
- Size of datatype can be changed. (\uparrow or \downarrow)

* Column should be empty for Decreasing the size or for changing the datatype from one to another

(which are incompatible)

Not necessary if it complies with existing values then okay

ALTER TABLE - QUIZ

VERY Important

Date: / /

Ques 5

(All the columns cannot be dropped)

ALTER TABLE Employee DROP COLUMN DOB,

=

DROP DOB;

* Some for ADD COLUMN = ADD

* Exercise

Fmp

→ * While Adding the new Column if it has to be the NOT NULL, then you have to provide the DEFAULT Value as well.

||

otherwise Error ~~X~~

DEFAULT: Can we give DEFAULT to PRIMARY KEY and UNIQUE.

* ALTER TABLE <table-name> ALTER COLUMN

Can only be used alter the data type or size of that column (not too incompatible manner)

||

No new name / No constraint / no Default with it.

* You can give Default with UNIQUE & Primary Key at the time of creation but later it may RAJDHANI constraint Violation.

3 ★ DROP TABLE

"Used to drop all the records as well as the structure of the table from the database"

Date: / /

* Syntax:

> **DROP TABLE <tablename>**

Note: Tables without foreign keys can be dropped directly.

* Exercise

* Exercise

* Exercise

* To Drop Parent Table

- * Remove the constraint
- * Drop child table first
- * Then Drop Parent table now
- * Drop Parent

without this if you try to delete the parent table → ~~X error~~ (integrity constraint violation)

* Child table can be deleted - easily (No Problem)

* Alternate Method to delete Parent Table

optional

> **DROP TABLE <tablename> IF EXISTS CASCADE;**
OR

> **DROP TABLE <tablename> IF EXISTS CASCADE;**

* → It drops the Parent table and removes the referential integrity or FOREIGN KEY constraint from the child table.

child table remains in the database

4# TRUNCATE

→ structural remarks as it is.

> DELETE from employees also works same
but Date: / /

TRUNCATE is faster here for this reason.

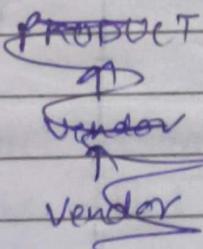
> TRUNCATE FROM Employee
~~TABLE~~

* Imp - we can't even DROP that column alone
that is being referenced. ✗ ↗

* We can't even update its value ← is no one reference

* we can't even delete that row ↓ if no one reference

* But we can Rename that column



I guess so

→ It can work for Multi-Level
Inheritance as WELL

DML!

CREATED CRUD

★ INSERT → "to add tuples/records to the table".

Date: / /

① Syntax:

Insert Statement 1

```
> INSERT INTO Employee
  VALUES
  ('1', 'Sandeep Potta', '2014-06-01')
    — All attributes
    — values
    |||
    [otherwise]
    X [caut error]
```

Insert Statement 2

```
> INSERT INTO Employee
  VALUES (Id, Fname, —
          all in
          any order.
          )
    — values in this order
```

⇒ for all the columns in
the order of their
specification during creation of table

Insert Statement 3

Select Query → Multiple rows can be inserted at a time through a single insert statement.

```
> Insert INTO Employee
  (Id, Fname, DOB) . . .
```

Select Query;

V-V-TMP

Points

→ 1) while inserting, the values should match with column datatype. (Don't put textual stuff into numeric)
* but we can if the stuff is totally numeric

→ 2) If column is String Character Date Time } Values must be in Single quotes ''

→ 3) String data is case sensitive and will be stored as given in the quotes "

* Double quotes not allowed

- For numeric data → no quotes
- NULL should not be enclosed in quotes otherwise treated as strng.

Date: / /

* Insertion when column names Not provided:

- i) put everything in VALUES in same order as specified while writing.

If NULL values are expected, then specify them using NULL

- ii) without columns in correct order.

↳

Datatype mismatch may happen → leads to Errors

↳ May accept → But human error → Business logic broken.

- iii) without column names - Missing values

↳ functn

→ count mismatch

* Insertion when column names are provided in

Insert statement

↳ If a value column name not mentioned

it will be added as NULL

↳

if acceptable turn OK → otherwise Error

- * Columns in correct order.

✓

- * In any order but should match with each other

- * If they do not match with each other → Error

↳ omitting NULL Attributes that allow NULL can be omitted from the insert statement (if NULL values needs to be inserted)

↳ omitting Default: Attributes that ~~not~~ has default value can be omitted even if NOT NULL constraint is there. (Default value is added)

- For numeric data → no quotes
- NULL should not be enclosed in quotes otherwise treated as string.

Date: / /

* Insertion when column names Not provided:

→ put everything in VALUES in same order as specified correctly.

If NULL values are explicitly expected, then specify them using NULL

2) Without columns in correct order.

↳

Datatype mismatch may happen → leads to Errors

↳ May accept → But human error → business logic broken.

3) Without column names - Missing values

↳ functions

→ count mismatch

* Insertion when column names are provided in

Insert Statement

↳ If a VALUES column name not mentioned
it will be added as NULL

↳

if acceptable then OK ⇒ otherwise Error

* Columns in correct order.

✓

* In any order but should match with each other

* If they do not match with each other → Error

* Omitting NULL: Attributes that allow NULL can be omitted from the insert statement (if NULL values needs to be inserted)

* Omitting Default: Attributes that ~~all~~ has default value can be omitted even if NOT NULL constraint is there (Default value is added)

* Insert Part -3

Inserting data when constraints are there on the columns

Date: / /

1) Primary Key Violation

If you can't insert the duplicate value for Primary key
you _____ NULL _____

2) Foreign Key Violation

If you can't insert the value for foreign key
that is not there in the parent table
referenced PK column

3) UNIQUE key constraint violation

If can't be added/ inserted duplicate
for unique attribute
constraint

4) NULLS are allowed for UNIQUE constraint
attribute.

Quiz - Insert Data :

Q1 ↳ Hayen Ko algaan se dekh.

↳ 2nd value me "Alice" → double quotes me ha!

Up X

Q2 ✓ ✗

Q3 ✗

After Alterations table ->

StudentId	NUMERIC
Name	NUMERIC
Gender	CHAR(1)

UPDATE

'to edit the data in rows'

[to increase salary by 10%]

Syntax

$$Sal \left(1 + \frac{10}{100} \right)$$

Date: 1/1/1

> UPDATE Employee SET Salary = Salary * 1.1;
[Change salary for all rows]

> UPDATE Employee SET Salary = Salary * 1.2 WHERE
only for rows where id = 1;
id = 1;

> UPDATE Employee SET Salary = Salary * 1.2, Bonus = 100;

[↓]
(update multiple ~~multiple~~ columns values)

*** Imp**

→ * No constraints must be violated
during the execution of an update
statement

* Any violation results in error.
(failure of update)

[* update without WHERE can be used to
update all the rows of the table]

1) Update Part 1: (Assuming no constraints are applied)

1) without where clause → updates all the rows.

2) Single column update →

↓

3) Multiple column update

4) Cannot update same value two times

X > update Employee SET Salary = 100, Salary = 200
where id = 1;

5) Multiple conditions in where.

Multiple rows will be updated.

Incorrect syntax → execute three employe

2* Update - Part 2

Date: / /

Integrity Violation

Constraint Violation

*1 NOT NULL Violation:

If a column has NOT NULL constraint

~~XX update Employee set Fname = NULL~~

"Error"

*2 PRIMARY can be updated except
in the constraints

Duplicates
and
NULL

*3 CHECK Constraint violation

keep in mind what values are allowed

*4 UNIQUE key violation

3* FOREIGN KEY violation :-> Update

Go see the examples Computer

↓
EmployeeId

(1) Foreign key violation - Child table.

~~↳ UPDATE Employee SET compId = 1005 WHERE empId = 1007;~~

integrity violation → compId not there in computer table

OR

(2) Foreign key violation - Master table :

→ Remove constraint
→ Do flags

if an id is referenced in the child

→ Add constraint again

then it cannot be updated / deleted

* Way to update in Parent Table

(1) First make that value in child table as NULL (No longer referencing)

(2) Now update that value in parent table

"so

(3) Now add that updated value back if you want

we can
delete / update

RAJDHANI

* The Referenced column by the Foreign Key must be the PRIMARY KEY or UNIQUE key of the Parent table.

Update - Quiz [Imp]

Q3 Tricky (

ID	Name	Gender	DOJ	marks	
2	Alan	F	1/2018-02-01	84	Updated
				(Q3)	updated
					Updated

DELETE : To remove one/more records from a relational database.

SYNTAX :

> DELETE FROM Employee; [Deletes all Rows]

> DELETE FROM Employee WHERE Dept = 'ETA'

Imp:

* Note Database system ensures that no constraints are violated during the execution of the DELETE statement.

* TRUNCATE (DDL) also removes all the records from the table
★ (→ Efficient and faster alternative to DELETE)

X Quiz Question - of update last → Go See

||| ★★★ FOR DELETE/UPDATE
RULES 2-3 Page
Pahit ke Padh
O Score. RAJDHANI

* DATE Comparison :

↳ uses $<=$, \geq , $<$, $>$ operators
and

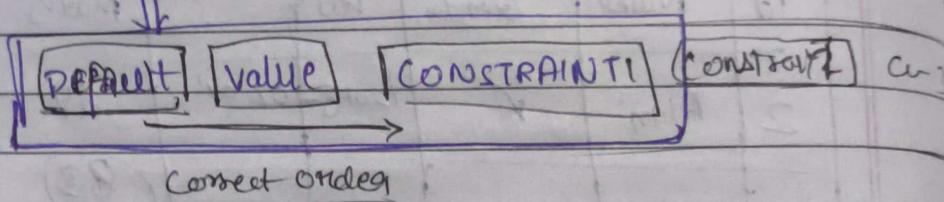
logical operators

Date: / /

→ Using LIKE '0004- -- - I - ;'

DBMS - nonneet - 02

* If column level constraint and Default both has to be given what must be the order??



* DROP can be used at the column level.

* TRUNCATE only at Table level

* TRUNCATE is faster.

IFNP : NOT NULL also be before any constraint

↓

because if we put CHECK constraint
before not null

and value comes out to be null
it will show error.