

DDL Command 2

Alter Table Command

→ To change the definition of a base table.

features

- Add a column. ✓
- remove a column. ✓
- change column definition. ✓
- Add or drop table constraints. ✓

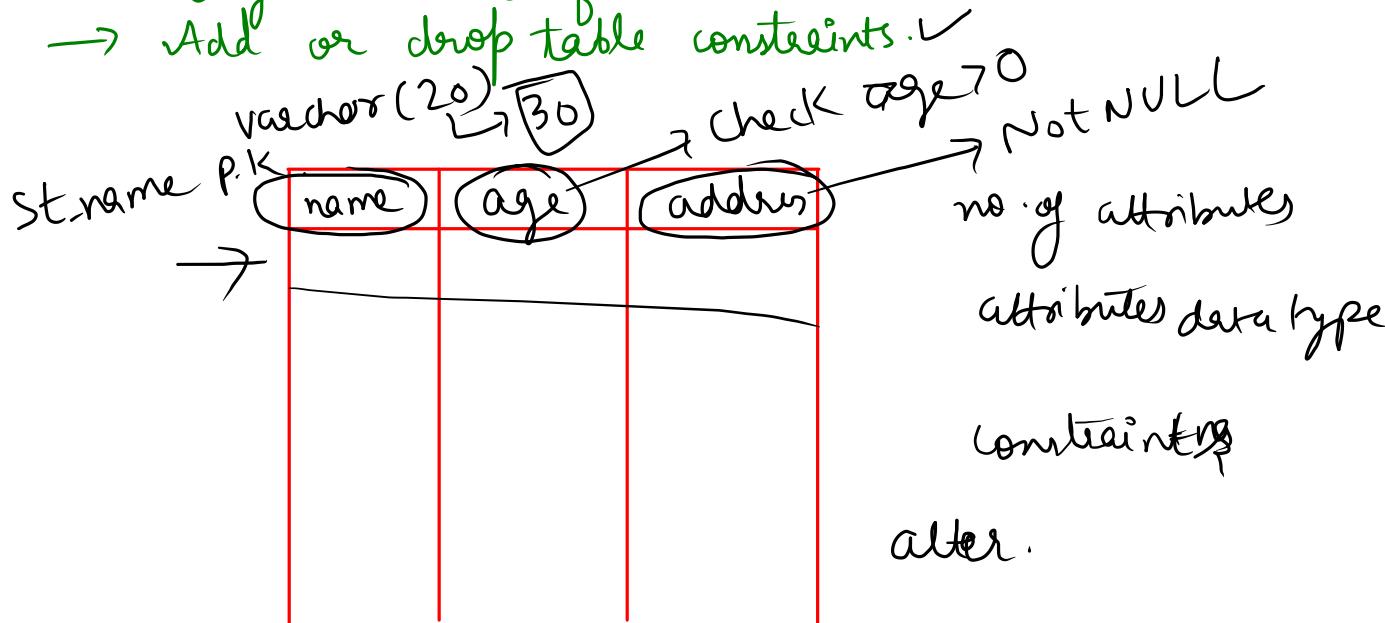


Table schema = Table

DDL commands -

~~Create~~
~~Alter~~
~~Drop~~
~~Truncate~~

DML.

~~Insert~~
~~update~~
~~delete~~.

Add a column

Alter table table-name add column-name datatype;

drop a column

Alter table table-name drop column column-name;

change name of column

Alter table table-name rename old-col-name to new-col-name;

change datatype of column

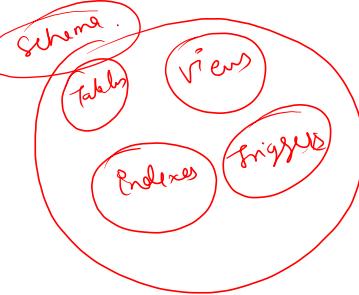
Alter table table-name alter column column-name type new-data-type;

Add a constraint

Alter table table-name add constraint constraint-name;

drop

Alter table table-name drop constraint constraint-name;



Drop table table-name

restrict(default)

Cascade

restrict

data present ✓

for F.K. Constraints X

all other constraints

data ✓

F.K. constraint ✓

all other constraints

drop

* elements → components used in definition of schema & NOT the data inside tables.

** default → If you have not mentioned explicitly Restrict/Cascade, then by default mode will be Restrict.

D.DL Command 3

Drop command ↗

→ If you try to drop a table, all the values present inside the table will be deleted, constraints present on that table will be deleted & then table too will be deleted.

Drop [schema | Table | constraint] <name> [Cascade | Restrict] ;

→ Cascade option will create a chain reaction and delete all the elements associated with element.

→ In restrict option, an element is deleted if it is not referenced in any constraints or views. otherwise, drop will not be executed & result in an error.

Truncate

Truncate table student;

Truncate command will delete all the data inside the table.

DDL Command 4

Truncate table table-name;

- Deletes all the values inside the table without altering or deleting schema of the table.

Qn:

Why is Truncate a DDL command & not DML?

Student

name	id	age
A	1	10
B	2	15
BCD	3	20

DML

age = 10
Delete From Student
where age % 10 == 0

A	1	10
BCD	3	20

Delete From Student =



Truncate can't be rolled back
Delete can be rolled back

DML Commands



- Insert ✓
- Update ✓
- Delete ✓
- Select ✓

Insert Command

student

name	id	age
abc	12	30

i) Insert into table-name values (v1, v2, v3, ...vn);

* here values should be provided in the order columns inside table are created while table creation.

ii) Insert into table-name (att1, att2, att3 ... - attn)
values (v1, v2, v3, ...vn);

* v1 value will be inserted against att1
v2 " " " " " att2
and so on.

Insert operation will result in error if

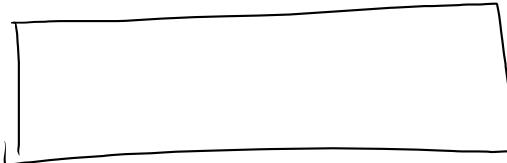
- i) datatype mismatch ✓
- ii) constraint violation ✓

* If values for certain columns are not provided while performing insert operation, then database will allocate default value corresponding to that column.

BLOB →

Insert into student values ('abc', 2, 30);

Insert into student (name, age, id)
values ('abc', 30, 12);

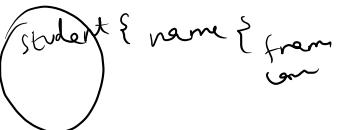


// Create table (

) With data (

select

)



Delete Command

→ It is used to remove tuples from a relation.

Syntax :-

Delete FROM table-name
Where column-name condition value ;

Update Command

- It is used to alter already present ^{imp.} **VALUES** inside a table.

Update table-name
 Set attr1=val1, attr2=val2, ... attrn=valn
 where attrK condition valueK.

- * Can throw an error if we have data type or constraint violation.

Update Student
 Set age = ~~id~~ * id * age
 Where ~~id > 2~~ = -10

	name	id	age			
→	A	1	10/10	10	10	10
→	B	2	20/20	10	40	✓
→	C	3	30/30	10	90	30
→	D	4	10/40	10	160	✓
→	E	5	50/20	10	280	50

Employee

Ename	ESSN	EDNO
A	1234	H
B	6789	M
-	-	-
-	-	-

Department

Parent		
DNUM	DName	DLOC
H	HR	Beth
-	-	-
-	-	-

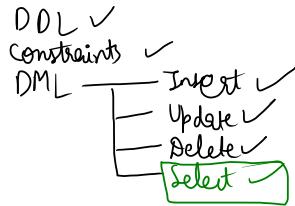
- i) Set null
- ii) Set default
- iii) Cascade

update cascade.
 cascade

foreign key (PAtt1) referenced tables (Att2) on
 delete

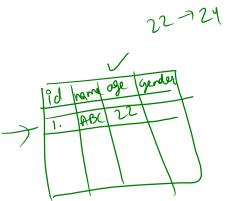
- i) set null
- ii) set default
- iii) cascade

SQL

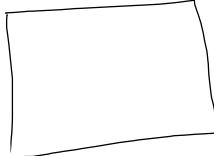


Agenda for today

Select from where
 aliasing
 operators
 pattern matching
 order by

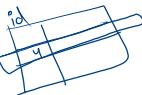


Select



Select Command

`SELECT attr1, attr2, attr3, ...
FROM tablename
WHERE attr1 condition attr2 ...;`



from
where
Select

Table name

	attr1	attr2	attr3	...	attrn
1	1	2	3	...	N
2	2	3	4	...	M
3	3	4	5	...	L
4	4	5	6	...	K
5	5	6	7	...	J
6	6	7	8	...	I
7	7	8	9	...	H
8	8	9	10	...	G
9	9	10	11	...	F
10	10	11	12	...	E
11	11	12	13	...	D
12	12	13	14	...	C
13	13	14	15	...	B
14	14	15	16	...	A

Select age, id, name
Select *

1. 'Select' and 'From' are mandatory, however 'where' is an optional clause.

2. Select * → This will output all the columns present in table.

3. Select *
From student, marks;
table1, table2 ...

→ Cartesian product.

Select department, number, dept location.

4. If 2 tables in join operation have same attribute names, then we use below

a) Select Student.name, Student.age, marks.name, marks.age
from Student, marks
where Student.name = marks.name and student.age = marks.age;

AND
OR
IN

b) Select S.name, S.age, M.name, M.age
From Student as S, Marks as M
where S.name = M.name AND S.age = M.age;

c) Select S.name, S.age, M.name, M.age
From Student S, Marks M
where S.name = M.name and S.age = M.age;

Aliasing is used to resolve ambiguity.

* Aliasing can be used in Select clause as well.

Select S.name as name_of_student, M.marks as marks_of_student
From Student S, Marks M
where S.name = M.name and S.age = M.age;

SQL treats table not as a set but as a multiset.

→ Duplicate entries are allowed in tables, views, query results.

⇒ An SQL table with a key is restricted to being a set.

Select DISTINCT keyword is used to eliminate duplicate tuples from the result set.

Select ALL keyword is used to select all the tuples from the result set.

Select All Salary
From Employee;

Select Distinct Salary
from Employee;

Select
From — —

Department

dname	dnumber	mgr-ssn	mgr-startdate
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

new -
data

Dept-locations

dnumber	dllocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Select *
from department,
dept_location;

Cross product with condition department.dnumber = dept_locations.dnumber

dname	dnumber	mgr-ssn	mgr-startdate	dnumber	dllocation
Research	5	333445555	1988-05-22	5	Bellaire
Research	5	333445555	1988-05-22	5	Sugarland
Research	5	333445555	1988-05-22	5	Houston
Administration	4	987654321	1995-01-01	4	Stafford
Headquarters	1	888665555	1981-06-19	1	Houston

Select * from
department, dept_locations
where dnumber = dnumber

dname	dnumber	mgr-ssn	mgr-startdate	dnumber	dllocation
Research	5	333445555	1988-05-22	1	Houston
Research	5	333445555	1988-05-22	4	Stafford
Research	5	333445555	1988-05-22	5	Bellaire
Research	5	333445555	1988-05-22	5	Sugarland
Research	5	333445555	1988-05-22	5	Houston
Administration	1	987654321	1995-01-01	1	Houston
Administration	4	987654321	1995-01-01	4	Stafford
Administration	4	987654321	1995-01-01	5	Bellaire
Administration	4	987654321	1995-01-01	5	Sugarland
Administration	4	987654321	1995-01-01	5	Houston
Headquarters	1	888665555	1981-06-19	1	Houston
Headquarters	1	888665555	1981-06-19	4	Stafford
Headquarters	1	888665555	1981-06-19	5	Bellaire
Headquarters	1	888665555	1981-06-19	5	Sugarland
Headquarters	1	888665555	1981-06-19	5	Houston

=
dept_location,
dept_location

Cartesian
product of
department and
dept_locations

where

Query 1

Retrieve the birth-date and address of all the employees whose name is 'John B Smith'.

Select bdate, address

From Employee

Where fname = 'John' and Minit = 'B'
And lname = 'Smith';

John john
 JOHN

Query 2

Retrieve the name and address of all the employees who work for the 'Research' department.

Select fname, address

from Employee, department

where Employee.dno = department.dnumber and
department.dname = 'Research';

Query 3

for every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address and birth date.

Query 4 for every employee retrieve the employee's first name and last name and the first & last name of his or her immediate supervisor.

Query 5: Retrieve the salary of every employee and all the distinct salary values.

~~Query 6.~~ Make a list of all project numbers for projects
that involve an employee whose last name is
'Smith', either as a worker or as a manager of
the department that controls the project.

Pattern matching operators

- It is done using 'LIKE' operator in where clause which is a comparison operator.
- It can be used for string pattern matching. Partial strings are specified using 2 reserved characters.
 - i) '%' \Rightarrow replaces 0 or more characters in a string.
 - ii) '_' \Rightarrow wildcard for a single character.

Query 7 Retrieve all employees whose address is in 'Houston TX'.

Query 8 find the employees who were born during 1950's.

~~Query 9~~

Show resulting salaries if every employee working on the 'ProductX' project is given a raise of 50%.

~~Query 10~~

Retrieve all employees in department 5 whose salary is between \$40,000 and \$50,000.