

DBT:

Constraints.

29/11/2021

- limitations/restrictions imposed on a table

EMP.

Emp No	Ename	Sal	Dept No.
1	A	5000	1
2	B	6000	1
3	C	7000	1
4	D	9000	2
5	E	8000	2

① Primary key constraint (primary column constraint)

- column or set of columns that uniquely identifies a row.
- duplicate values are not allowed (has to be unique)
- Null values are not allowed (is a mandatory).
- it's recommended that every table should have a primary key.
- It helps from a long term perspective.
- The purpose of primary key is row uniqueness (with the help of primary key you can distinguish between 2 rows of a table).
- rowid is pseudo column so we can not make it as primary key.
- rowid is encrypted so we can't use
- rowid is not constant. if we update any row, the row id may change. so to keep track of changed rowid is difficult task for user, so don't make rowid as primary key.
- text and blob cannot be primary key.
- when we declare any column as primary key, unique index is automatically created.

② Composite primary key :-

- combine 2 or more columns together to serve the purpose of primary key.

Snehal Sawant

- can combine upto 32 columns in a composite primary key
- if you create / have a composite primary key, then composite ~~key~~ ^{key} uniqueness is automatically created.
- you can have only 1 primary key constraint per table.
- Candidate key → is not a constraint ex. panno, Pass postNo
- Candidate key → is a definition.
- Candidate key → bcs

Steps for identifying primary key: -

1. Key element will be primary key of your table.
 2. If you cannot identify some element, then try for composite primary key.
 3. If you cannot identify composite primary key, then add an extra column to the table to serve the purpose of primary key.
- Surrogate key → is not a constraint
 - Surrogate key → is a definition
 - Surrogate key → if you add an extra column to the table to serve the purpose of primary key then such a primary key is known as surrogate key (then char data type is recommended)

• Create table emp

```
( empno char(4) primary key,
  ename varchar(25),
  sal float,
  deptno int
);
```

Snehal Sawant

- insert into emp values ('5', 'F', 5000, 2); ← Error.
- all constraints are at server level (you may perform the DML operations using any front-end s/w, the constraints will always be valid).

- internally a constraint is a mysql created function; it performs the validations.
 - select * from information - schema . table - constraints .
 - select * from information - schema . table - constraints where table - schema = 'cdacmumbai';
 - select * from information - schema . key - column - usage where table - name = 'emp';
 - show indexes from emp;
 - To drop the constraint:
 - alter table emp drop primary key;
 - To add the primary key afterwards to an existing table:-
 - alter table emp add primary key (empno);
 - For composite primary key:
- ```

create table emp
(
 empno char(4)
 ename varchar(25),
 sal float,
 deptno int,
 primary key (deptno, empno)
);

```

# Constraints are of 2 types:-

1. Column level constraint (specified on 1 column)
2. Table level constraint (specified on 2 or more column n) (composite)  
(has to be specified at the end of the structure).

- alter table emp add primary key (deptno, empno);

*Snehal Sawant*



## # Not Null:-

- Null values are not allowed (is a mandatory column) (similar to PK)
- duplicate values are allowed (~~unlike~~ unlike PK)
- you can have any number of not null constraints per table (unlike PK)
- you cannot have a composite not null constraint (unlike PK)
- always a column level constraint.

• Create table emp

```
(empno char(4),
 ename varchar(25) not null,
 sal float not null,
 deptno int
);
```

*Snehal Sawant*

• To see not null column.

```
desc emp;
```

• To add the not null constraint after wards to an existing table;

```
alter table emp modify ename varchar(25) not null;
```

• To drop the not null constraint:-

```
alter table emp modify ename varchar(25) null;
```

• solution for candidate key columns:-  
not null constraint + unique index.

• with the help of above, you can 'indirectly' have multiple primary key constraints in a table.

• Alternate key → is not a constraint

Alternate key → is a definition

Alternate key → for the candidate key column, if you specify a not null constraint and you create an unique index, then it becomes an alternative



to primary key; then such a candidate key is known as alternate key.

- super key → is not a constraint
- super key → is a definition
- super key → if you have an alternate key in the table then the primary key column is known as superkey.

# Unique:

- duplicate are not allowed (similar to PK).
- null values are allowed. (you can specify any number of null values).
- text and blob cannot be unique.
- unique index is created automatically.
- you can combine upto 32 columns in a composite unique.
- you can have any number of unique constraints per table (unlike PK).

• Create table emp

```
(empno char(4),
 ename varchar(25),
 sal float,
 dept no int,
 mob-no char(15) unique,
 unique (deptno, empno)
);
```

*Snehal Sawant*

← column level const.

← table level const.

• to drop the unique constraint :-

```
drop index mob-no on emp;
```

```
drop index dept-no on emp;
```

• to add constrain afterwards :-

```
alter table emp add unique (mob-no);
```

or

```
alter table emp add constraint u-emp-mob-no
unique (mob-no);
```



create table emp

```
(
 empno char(4)
 ename varchar(25),
 sal float,
 deptno int,
 mob_no char(15),
 unique (deptno, empno),
 unique (mob-no)
);
```

- Column level constraint can be specified at table level (at the end of the structure), but a table level composite constraint can never be specified at column level.
- Column level constraint can be specified at table level (at the end of the structure), except for the not null constraint, which is always a column level constraint, and therefore the syntax will not supported specifying it at table level (at the end of structure).

# ~~Foreign table: key~~

FOREIGN KEY (foreign column)  
↓  
child column (F.K.)

EMP.

| EMPNO | ENAME | SAL  | DEPTNO | MGR |
|-------|-------|------|--------|-----|
| 1     | A     | 5000 | 1      | 1   |
| 2     | B     | 6000 | 1      | 1   |
| 3     | C     | 7000 | 1      | 1   |
| 4     | D     | 8000 | 2      | 2   |
| 5     | E     | 9000 | 2      | 2   |
| 6     | F     | 9000 | 2      | 2   |

DEPT.

| DEPT NO | DNAME | LOC |
|---------|-------|-----|
| 1       | TRN   | Bby |
| 2       | EXP   | DLH |
| 3       | MKTG  | Cal |

→  
Parent  
column.  
(P.K.)

Snehal Sawant



- Column that has been derived from elsewhere.
- Column or set of columns that references a column or set of columns ~~that references a column or set of~~ some table.
- Foreign key constraint is specified on child column (not the parent column).
- Parent column has to be primary key or unique (this is pre-requisite for foreign key).
- We can have null value in primary key w.r.t. f.k.
- Foreign key (child column) will allow duplicate values (unless specified otherwise).
- Foreign key (child column) will allow null values (unless specified otherwise).
- Text and blob cannot be foreign key.
- Index for child column is not created automatically over here. (If you want the index then you will have to create it manually).
- Foreign key (child column) may reference column of same table (known as self referencing).

• Create <sup>table</sup> dept  
 ( deptno int primary key, ← parent table  
 dname varchar(15),  
 loc varchar(10)  
 );

Create table emp

( empno char(4) primary key, ← child table  
 ename varchar(25),  
 sal float,  
 deptno int,  
 mgr char(5),  
 constraint fk-emp-deptno foreign key (deptno)  
 references dept(deptno),  
 constraint fk-emp-mgr foreign key (mgr)  
 references emp(empno) )

*Snehal Sawant*



- Constraint fk\_emp\_deptno → optional
- Constraint fk\_emp\_mgr → optional.

• To drop the constraint:

alter table emp drop foreign key fk\_emp\_deptno;

• if you want the constraint afterwards to an existing table:-

alter table emp  
add constraint fk\_emp\_deptno  
foreign key (deptno)  
references dept (deptno);

*Snehal Sawant*

- you can delete the parent row provided child rows don't exist.
- you cannot delete the parent row when child rows exist.

- 1) delete from emp where deptno = 2;
- 2) delete from dept where deptno = 2;

On delete cascade:- if you delete the parent row then MySQL will automatically delete child rows also

# Make change while creating table.

- Constraint fk\_emp\_deptno foreign key (deptno)  
references dept (deptno) on delete cascade;
- remaining create statement will be same.

• to preserve the child rows:-

- 1) update emp set deptno = null where deptno = 2;
- 2) delete from dept where deptno = 2;

- update dept set deptno = 4 where deptno = 3;

• you can update the parent column provided the child rows don't exist.



- you cannot update the parent column when the child row exist.
- update dept set deptno = 4 where deptno = 2; Not allowed.

### on update cascade →

if you update the parent column then MySQL will update the child rows also

- constraint fk-emp-deptno foreign key (deptno) references dept(deptno) on delete cascade on update cascade;

- insert into emp values ('7', 'G', 7000, 2, '7');
- first it inserts, then it checks for the constraint, it will allow.

- insert into emp values ('7', 'G', 7000, 2, '8'); ← Error;
- first it inserts, then it checks for the constraint, then it will rollback, and gives an error message.

### Emp 2.

| EmpNo | Ename | MGR. |
|-------|-------|------|
| 7     | ---   | 10   |
| 8     | ---   | 9    |
| 9     | ---   | 8    |
| 10    | ---   | 7.   |

if Emp 2

| Emp No | Ename | MGR. |
|--------|-------|------|
| 7      | ---   | 10   |
| 8      | ---   | 9    |
| 9      | ---   | 8    |
| 10     | ---   | 11?  |

- insert into emp (select \* from emp2);

- insert into emp (select \* from emp2);

Error

11 is not present it will rollback.

*Snehal Sawant*



on delete cascade,

- delete from emp where empno = 1;
- all the rows will delete
- avoid on

Assumption on delete cascade for mgr & Empno:-

- avoid on delete cascade in the event of self-referencing, you may delete more rows than expected.
- it's safer to use on delete cascade across 2 tables.

# Check constraint.

- use for validation s (used for checking purposes)
- e.g. dele date > = order date, age > 21, etc.

• ~~Create table emp~~

- we can use Relational operators, special operators, Logical operators, Arithmetic operator, call single row fun eg. upper, lower, etc.

• default is not a constraint

• default is a clause that you can use with create table

• if you specify some value, then it will take that value, if nothing is specified, then it will take default value.

• to make use of default value - use following insert statement.

insert into emp (ename, dept no, comm, mob\_no)  
values (--- -- -- --);



Create table emp

```
(empno int auto-increment primary key,
 ename varchar(25) check (ename = upper (ename)),
 sal float default 7000
 check (sal between 5001 and 199999),
 deptno int,
 status char(1) default 'T'
 check (status in ('T', 'P', 'R')),
 comm float not null,
 mob_no char(15) unique,
 check (sal + comm < 3000000),
 constraint fk_emp_deptno (deptno) references dept
 (deptno)
);
```

STATUS.

T → ~~Training~~ temporary

P → Permanent

R → Retired.