CAP -> consistency, availability, partition
↓
sql
↓
Big data

24/5/19

Data manipulation language (DML)

it is used to manipulate the data from table

Insert

=> Insert into tableName (column1, column2 ....);
values (value1, value2 ....)

=> Insert into tableName
values (value1, value2, value3 ...);

- Describe emp;

sql> desc emp;

sql> insert into emp(empno, ename, mgr, sal, hiredate, job, deptno)
values (1234, 'varsha', 456, 35000, sysdate, 'pm', 40);

sql> Insert (4567, 'ABHI', null, null, sysdate, 1500, null, null);

Errors

sql> insert (4567, 'ABHI, null, null, sysdate, 1500, null, null);
unique constraint

sql> insert (null, 'ABHI', null, null, sysdate, 1500, null, null
cannot insert null

update

sql> update emp set sal = 1000
where empno = 7369; -> only specified empno will updated

sql> update emp set sal = 1000
where job = 'salesman'; -> same here as above
-> rows

sql> update Emp set SAL = 1000; -> all rows

=> update all
sql> update emp
where ...

sql> Roll Back;

sql> select * from

sql> Delete from
1 row

sql> Delete from

sql> Roll back
roll back

* the ... which ...
* all the ... either ... statement.
* changes ... to make ... ...

Transaction
1) commit
the change...
2) Roll Back:-
to changes...
commit
3) save point
...

⇒ update all the employee salary w.r.t to manager salary

SQL> update emp set sal = (select sal from emp mgr

where emp.mgr = mgr.empno);

SQL> roll back;

Delete

SQL> select * from emp;

SQL> Delete from emp where empno = 7369;

1 row deleted

SQL> Delete from emp;

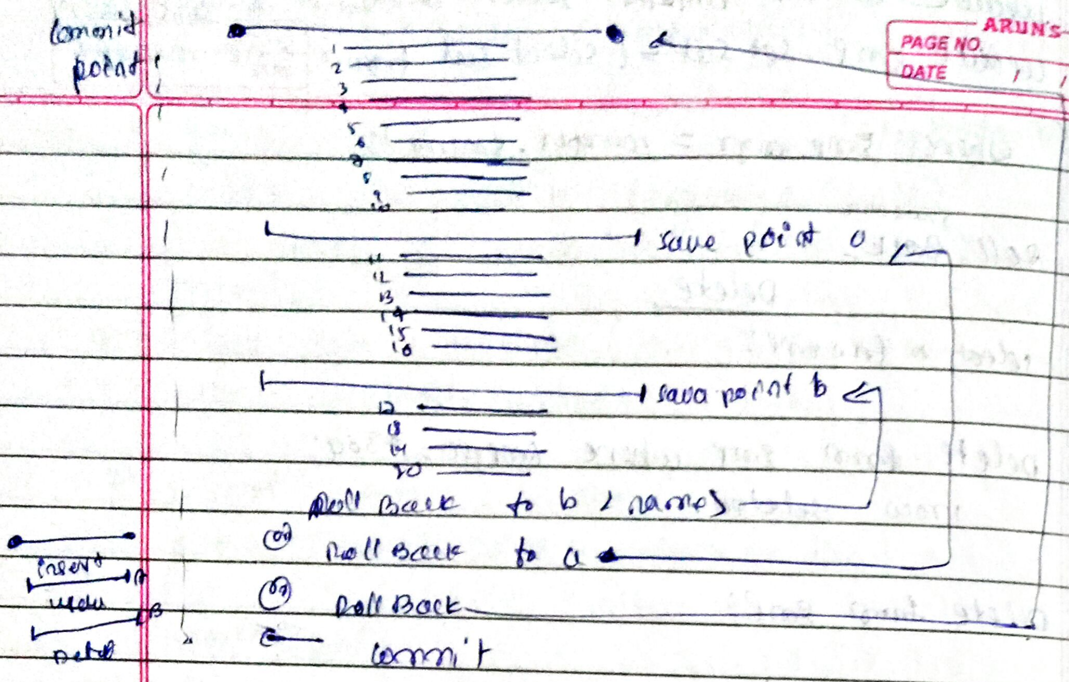
SQL> Roll back;

roll back complete.

* the out come of any DML statement is integer values which represent no of records affected.
* all the statement in SQL follows transaction but it is either the statement executed completly or not executed statement.
* Changes made by DML statement is not permanent, hence to make it permanent transaction control statement is used

Transaction

1) commit :- commit statement is used to make the changes in the database permanent.
2) Roll Back :- Roll Back statement is used to undo (reverse) the changes made on to the database either to previous commit point or to specified save point
3) save point :- save points are intermediate temporary points in a transaction can be rolled back



27/08/18

Data Definition language

which is used to define the data

SQL> select table-name from all-tables where table name like 'E%';

SQL> select table-name from user-tables;

Table_name

Dept, Emp, Bonus ...

SQL> select table_name from all-tables;

create table doctor
(columnname datatype constraints
column name2 datatype constraints
;
);

SQL> create table doctor
(d_id number(4) primary key,
name varchar2(20) not null,
doc_type varchar2(10),
exp number(2) check(exp>5),
phno number(10) check(length(phno)=10),

email varch...
);
Table c...

SQL> desc doctor
name
D-ID Nam...

SQL> select * ...
no rows ...

SQL> create ta...
(P-id numb...
name varch...
age num...
gender char...
t-date d...
d-id number...
addr varch...
phno number...
);

SQL> rename doc...

SQL> desc doctor ;
Error

SQL> desc doc;
name -> ...

SQL> select create ...
Tab...

SQL> create tabl...
Table cr...

SQL> desc Emp3 ...
Emp to ...

SQL> select ...
no r...

email varchar(15) check(email like '%@gmail.com'));

Table created

SQL> desc doctor
Name
D_ID NAME DOC_TYPE - - -

SQL> select * from doctor
no row selected

SQL> create table patient
(p_id number(4) primary key,
name varchar(20) not null,
age number(3) not null,
gender character(1) not null check(gender in('M','F')),
t_date date default(sysdate),
d_id number(4) references doctor(d_id),
addr varchar(30) default('Bangalore'),
phno number check(length(phno)=10) unique not null
);

SQL> rename doctor to doc;

SQL> desc doctor;
ERROR

SQL> desc doc;
Name -> D_ID. NAME DOC_TYPE - - -

SQL> create table emp2 as (select * from emp);
Table created

SQL> create table emp3 as (select * from emp where 1=2);
Table created. -> In this query only structure will copy

SQL> desc emp3;
Emp to Name - - - -

SQL> select * from emp3;
no row selected

SQL> select table_name from user_tables;
Table names

ARUN'S
PAGE NO.
DATE 28/5/19

Drop table :- select * from emp2;
SQL> select * from emp3;
no rows selected.

SQL> drop table emp3;
Table dropped.

SQL> select * from emp3;
error

*Once the table is dropped it is stored in recyclebin in SQL
Show recyclebin will display the contents of recycle bin.
SQL> show recyclebin;

To delete the table from the recycle bin permanently use
temporary time
purge table tablename.
SQL> purge table emp3;

To delete all the content of the recyclebin the command is
SQL> purge recyclebin;

To restore the deleted table from the recyclebin (forward)
syntax: Flashback table tablename to before drop;
SQL> Flashback table emp1 to before drop;

To delete permanently without saving in recycle bin
syntax: drop table tablename purge;
SQL> drop table emp1 purge;

truncate
truncate table table name;
SQL> truncate table emp2;

SQL> desc emp2;

SQL> select * from
no rows sele

SQL> roll back;

SQL> select * from
no rows sele

By using truncate
data structure
records are
deleted.

Alter stat
a) Adding col
SQL> alter ta
add colu

SQL> alter
add

b) To rename
Syntax:- Alter ta
rename
SQL> Alter tab
rename

c) modifying
syntax:- alter
modify

SQL> alter
modify

d) To dr
Syntax:- Alter
drop

SQL> alter
drop

SQL> select * from emp2;
no rows selected

SQL> roll back;

SQL> select * from emp2;
no rows selected.

By using truncate we can delete all the records but the data structure remains same. But once truncate the records we cannot restore records and it will permanently deleted.

Alter statement

a) Adding column

Syntax> alter table tablename
add columnname Datatype Constraint

SQL> alter table emp2
add phno number(10) check (length(phno)=10);

b) To rename the column

Syntax:- Alter table tablename
rename column oldcolumnname to new columnname.

SQL> Alter table emp2
rename column phno to mob no;

c) modifying the column (changing the data type)

Syntax:- alter table tablename
modify columnname Datatype

SQL> alter table ~~table~~ emp2
modify sal number(10,2);

only we can change same data type of column to be modified must be empty to change data type

d) To drop a column

Syntax:- Alter table tablename
drop column column name

SQL> alter table emp2
drop column mobno;

Sqcs desc emp2;

SQL> alter table dept
drop column deptno;
cannot drop parent key column

el adding & removing not null constraints
SQL> Alter table emp2
modify empno null;

SQL> Alter table emp2
modify empno not null & null;

User constraints

User constraints are the constraints applied on the table by the user
* In DB the user constraints are identified by constraints name
* user constraints table is the table in DB which stores the details of the constraints created by the user.

user constraints
SQL> desc user_constraints
user constraints are the constraints on the table by user
SQL> select constraint_NAME, CONSTRAINT_TYPE, table_NAME from user_constraints;

ARJUN'S
PAGE NO.
DATE / /

To unlock other accounts

```
Sql> conn system/Tiger
connected
Sql> alter user HR identified by tiger account unlock;
user altered
SQL> conn hr/Tiger
connected
```

Normalization

Normalization is a process of achieving redundancy free, anomalies free table which helps in storing the data in a constant manner

* unnecessary duplicate data is called redundancy
* Anomaly such as insertion anomaly, update anomaly & deletion anomaly is abnormalities during the respective operations.

USN	name	dept	deptno	dph
CS01	AKON	CS	10	10001
EC01	B KON	EC	20	10002
ME01	C KON	ME	30	10003
EC01	D KON	EC	20	10002

* Insertion Anomaly :- same data as dept, dept no, depth has to be inserted again and again

* update Anomally :-

* delete Anomally :-

* normalization is also helps in distributing the data across multiple tables & defines relationship across
* Normalization is achieved by applying certain rules given as Normal forms

USN	name	subcode	sub	marks	deptno	dept
EC001	Akon	FT01, SS01	FT, SS	28, 25	10	EC
CS001	Bkon	SE01, OS01	SE, OS	22, 17	20	CS
EC002	Ckon	SS01, NA02	SS, NA	18, 22	10	EC
ME001	dkon	BT D01	BTD	27	30	ME
CS002	ekon	SE01, M102	SE, MC	12, 06	20	CS

Normal Form

ARUN'S
PAGE NO.
DATE / /

* sequence of records & fields does not matter
* field name should be unique in a table
* every field should have values that belongs to some domain.
* every field should be a single (Atomic value) unique valued attribute.

USN	name	sub code	subs	marks	DeptNo	Dept
EC001	AKON	FT01	FT	88	10	EC
EC001	AKON	SS01	SS	85	10	EC
CS001	BKON	SE01	SE	22	20	CS
CS001	BKON	OS01	OS	17	10	CS
EC002	CKON	SS01	SS	19	10	EC
EC002	CKON	MA02	MA	22	10	EC
MT001	dKON	BTP04	BTP	87	30	MT
CS002	EKON	SE01	SE	12	20	CS
CS002	EKON	MP02	MP	06	20	CS

II Normal Form

* Table should be in I normal form
* there should no partial functional dependency

Function dependency :-

if a column is functionally dependent on another column for it's value

eg :- subject, sub code
dept no, dept

composite primary key

if 2 or more columns is involved in defining a primary key, then it is called as composite primary key

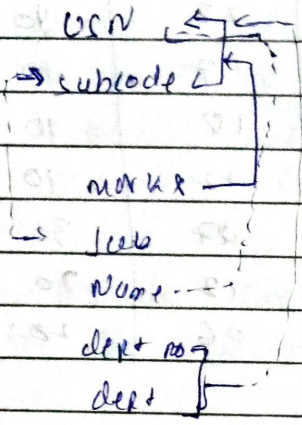
ARUN'S
PAGE NO.
DATE / /

Fully functional Dependency :-

If a column is functionally dependent on all the column of composite primary key, then it known as fully functional dependency

partial functional

If a column is functionally dependent on some of the column of composite primary key but not all, then it is known as partially functional dependency.



CRC

F.K.

USN	Name	dept no	dept
EC001	Akon	10	EC
CS001	Bkon	20	CS
EC002	Ckon	10	EC
ME001	Dkon	30	ME
CS002	Ekon	20	CS

Scub	subcode
ET	FT01
CS	SS01
SE	SE01
OS	OS01
NA	NA02
BD	BT001
NL	NL02

F.K
USN
EC001
EC001
CS001
CS001
EC002
EC002
ME001
CS002
CS002

II

F.K F.K

USN	SC	Marks
EC001	FT01	28
EC001	SS01	25
CS001	IE01	22
CS001	OS01	17
EC002	SS01	18
EC002	NA02	22
AE001	BTP01	27
CS002	SE01	12
CS002	MP02	6

F.K P.K

USN	name	Dept-no

Dept no	Dept name

III NF

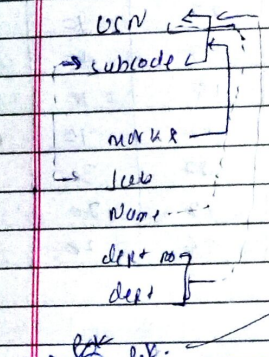
III rd NF

Fully functional dependency :

It is a column is functionally dependent on all the column of composite primary key, then it is known as fully functional dependency

partial functional

It is a column is functionally dependent on some of the column of composite primary key but not all then it is known as partially functional dependency



CRIC

P.K.

USN	Name	Dept no	Dept
EC001	Akon	10	EC
CS001	Bkon	20	CS
EC002	Ckon	10	EC
ME001	Dkon	30	ME
CS002	Ekon	20	CS

Sub	Subcode
ET	ET01
CS	SS01
SE	SE01
OS	OS01
NA	NA02
BD	BT001
ME	ME02

F.K

USN	SC	Marks
EC001	ET01	28
EC001	CS01	25
CS001	EC01	22
CS001	OS01	17
EC002	SS01	18
EC002	NA02	27
ME001	BT001	27
CS002	SE01	17
CS002	NA02	6

FK

USN	Name	Dept no

PK

Dept no	Dept name

II NF

II nd NF