CS245: Databases

SQL

Vijaya saradhi

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati

Data inserting/updating/deletion

- Inserting rows into table
    - One row
        - Insert all the columns of the row
        - Inserting fewer columns of the row
    - DEFAULT columns cases
    - Two rows
    - Loading a local file
- Updating rows in the table
    - One row
    - Multiple rows
- Deleting rows from the table
    - One row
    - Multiple rows

Insert one row

Inserting one row

- Insert all the columns of the row
- Inserting fewer columns of the row
- Specify table into which the row will be inserted
- Is the row added at the beginning? in the middle? or at the end?

R				
c1	c2	c3	c4	c5
1	2	3	4	5

```
INSERT INTO R(c1, c2, c3, c4, c5) VALUES (1, 2, 3, 4, 5);
```

Insert one row

Inserting one row

R				
c1	c2	c3	c4	c5
1 10	2 20	3 30	4 40	5 50

```
INSERT INTO R(c1, c2, c3, c4, c5) VALUES (10, 20, 30, 40, 50);
```

Insert one row - specify few columns

All columns having no constraints

R				
c1	c2	c3	c4	c5
1	2	3	4	5
10	20	30	40	50
100	$\perp$	300	400	$\perp$

```
INSERT INTO R(c1, c3, c4) VALUES (100, 300, 400);
```

Insert one row - specify few columns

c2 cannot take NULL values

say c2 has NOT NULL constraint
constraint violation: INSERT statement is rejected by DBMS

R				
c1	c2	c3	c4	c5
1	2	3	4	5
10	20	30	40	50
100	⊥	300	400	⊥

```
INSERT INTO R(c1, c3, c4) VALUES (15, 35, 45);
```

Insert one row - specify few columns

DEFAULT value constraint

say c2 has DEFAULT value constraint as 250
while inserting, only c1, c3 & c4 values are being inserted, due to default constraint on column c2, 250 also insert along with c1 = 150, c3 = 350, c4 = 450

R				
c1	c2	c3	c4	c5
1	2	3	4	5
10	20	30	40	50
100	⊥	300	400	⊥
150	250	350	450	⊥

```
INSERT INTO R(c1, c3, c4) VALUES (150, 350, 450);
```

Insert one row - specify few columns

FOREIGN KEY constraint

say c2 is a foreign key pointing to cid of table S
Table S do not have cid=22 (c2)
INSERT statement will be rejected by DBMS

R				
c1	c2	c3	c4	c5
1	2	3	4	5
10	20	30	40	50
100	200	300	400	⊥
150	250	350	450	⊥

S		
cid	cname	cedits
2	SQL	3
20	C++	6
200	R	4
250	Python	8

```
INSERT INTO R(c1, c2, c3, c4, c5) VALUES (11, 22, 33, 44, 55);
```

Insert two rows

Inserting two rows

R				
c1	c2	c3	c4	c5
1	2	3	4	5
10	20	30	40	50
100	⊥	300	400	⊥
150	250	350	450	⊥
170	270	370	470	570
180	280	380	480	580

```
INSERT INTO R(c1, c2, c3, c4, c5) VALUES (170, 270, 370, 470, 570), (180, 280, 380, 480, 580);
```

Insert a local file into a table

File must meet all table constraints

Invoke mysql as: `mysql -uroot -p --local-infile`
to read data from local files

R				
c1	c2	c3	c4	c5

```
LOAD DATA LOCAL INFILE '/home/saradhi/tableR-data.csv'
INTO TABLE R
FILEDS TERMINATED BY ','
LINES TERMINATED BY '\n';
```

Insert a local file into a table

File must meet all table constraints

First line of the file contains header; ignore header

R				
c1	c2	c3	c4	c5

```
LOAD DATA LOCAL INFILE '/home/saradhi/tableR-data.csv'
INTO TABLE R
FILEDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

Insert a local file into a table

File must meet all table constraints

First 10 lines of the file contains header and comments; ignore them

R				
c1	c2	c3	c4	c5

```
LOAD DATA LOCAL INFILE '/home/saradhi/tableR-data.csv'
INTO TABLE R
 FILEDS TERMINATED BY ','
 LINES TERMINATED BY '\n'
IGNORE 10 LINES;
```

Insert a local file into a table

File must meet all table constraints

Columns are separated by space

R				
c1	c2	c3	c4	c5

```
LOAD DATA LOCAL INFILE '/home/saradhi/tableR-data.csv'
INTO TABLE R
 FILEDS TERMINATED BY '␣'
 LINES TERMINATED BY '\n'
IGNORE 10 LINES;
```

Insert a local file into a table

File must meet all table constraints

Columns are separated by '#'

R				
c1	c2	c3	c4	c5

```
LOAD DATA LOCAL INFILE '/home/saradhi/tableR-data.csv'
INTO TABLE R
 FILEDS TERMINATED BY '#'
 LINES TERMINATED BY '\n'
IGNORE 10 LINES;
```

Specifying order of row insertion?

Can we instruct DBMS?

- Row storage is internal to the DBMS
- This burden of storage is decoupled from users
- A table with primary key constraint, records are stored in the sorted order of the primary key
- Detailed discussion of storage will be covered when discussing DBMS internals

Updating one row

Updating one row

Assume c1 is a primary key column

R				
c1	c2	c3	c4	c5
1	2	3	4	5
10	20	30	40	50
100	$\perp$	300	400	$\perp$
150	250	350	450	$\perp$

This update statement will be allowed

UPDATE R **SET** c1 = 5 **where** c1 = 1;

Updating one row

Updating one row

Assume c1 is a primary key column

R				
c1	c2	c3	c4	c5
1	2	3	4	5
10	20	30	40	50
100	⊥	300	400	⊥
150	250	350	450	⊥

This update statement will be allowed

UPDATE R **SET** c1 = 5 **where** c1 = 1;

This update statement will be rejected

UPDATE R **SET** c1 = 10 **where** c1 = 1;

Updating multiple rows

Updating multiple rows

R				
c1	c2	c3	c4	c5
1	2	3	4	5
10	20	30	40	50
100	$\perp$	300	400	$\perp$
150	250	350	450	$\perp$

```
UPDATE R SET c1 = 101 where c1 >= 100;
```

Deleting one row

Deleting one row

Assume c1 is a primary key column

R				
c1	c2	c3	c4	c5
1	2	3	4	5
10	20	30	40	50
100	⊥	300	400	⊥
150	250	350	450	⊥

```
DELETE FROM R WHERE c1 = 1;
```

Deleting multiple rows

Deleting multiple rows

R				
c1	c2	c3	c4	c5
1	2	3	4	5
10	20	30	40	50
100	$\perp$	300	400	$\perp$
150	250	350	450	$\perp$

```
DELETE FROM R WHERE c1 >= 100;
```