

Key Points – Data Modification

- GUI tools like MySQL Workbench may provide interfaces for inserting, modifying, and deleting data but knowing the SQL is important for developing DB applications

Data Modification

- insert rows into a table
 - insert a single row
`INSERT INTO R(A1,...,An)`
`VALUES (v1,...,vn)`
 - uses specified values for attributes listed and default values for the others
 - insert multiple rows
`INSERT INTO R(A1,...,An)`
`<subquery>`
 - subquery is fully evaluated before any rows are inserted, so no inserted rows can affect the subquery

LOAD DATA INFILE

- bulk loads rows from a text file into a table
 - faster than doing individual INSERTs
- LOCAL means the file is on the client machine and is sent to the server
 - without it, the file must be on the server host
- local loading must be enabled on both server and client
- clauses must appear in order: FIELDS, LINES, IGNORE, column list, SET

```
loan_id,borrower_id,copy_id,checkout_date,due_date
1,101,5001,2026-10-01,2026-10-15
2,102,5002,2026-10-03,2026-10-17
3,101,5007,2026-10-05,2026-10-19
```

```
LOAD DATA LOCAL INFILE '/home/user/cs343/loans.csv'
INTO TABLE LOAN
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(loan_id, borrower_id, copy_id, checkout_date, due_date);
```

LOAD DATA INFILE

- delimiters – FIELDS TERMINATED BY
 - tab ('`\t`') is the default
- quoted fields
 - ENCLOSED BY '`''`' – every field is quoted
 - OPTIONALLY ENCLOSED BY '`''`' – only some fields are quoted
 - common for CSV, where values containing commas are quoted so they will load correctly
 - a doubled quote inside a quoted field ("`5'' shelf`") loads as one quote character
- line endings – LINES TERMINATED BY
 - '`\n`' for Unix/Mac (default), '`\r\n`' for files created on Windows
 - using the default for Windows files means a stray `\r` at the end of the last column
- header rows
 - IGNORE `n` LINES skips the first `n` lines
 - e.g. IGNORE 1 LINES for a column name header row

LOAD DATA INFILE

- column list – maps file columns to table columns in file order
 - omit table columns not present in the data file
 - will get its DEFAULT value if there is one, a generated value if it is AUTO_INCREMENT, or NULL if it is nullable with no default
 - skip unwanted file columns by assigning to a throwaway variable e.g. (borrower_id, @dummy, copy_id, checkout_date, due_date)
- SET
 - @var in the column list captures a value for transformation
 - e.g. to convert dates not in YYYY-MM-DD or to turn empty strings into NULLs

```
LOAD DATA LOCAL INFILE '/home/user/cs343/borrowers.csv'  
INTO TABLE BORROWER
```

```
...  
(borrower_id, name, email, @phone, @join_date)  
SET phone = NULLIF(@phone, ''),  
    join_date = STR_TO_DATE(@join_date, '%m/%d/%Y');  
...
```

turns empty phone into NULL
turns MM/DD/YYYY date into YYYY-MM-DD

LOAD DATA INFILE

- dates must be in YYYY-MM-DD format unless converted with STR_TO_DATE
- \N in the file loads as NULL; an empty field loads as an empty string or 0
- use forward slashes in Windows paths or double the backslashes
- with LOCAL, rows that violate a unique key or have bad data are skipping with warnings instead of stopping the load

Data Modification

- delete rows from a table

```
DELETE  
FROM R  
WHERE <condition>
```

 - all rows satisfying the condition are deleted
 - without WHERE, all rows are deleted (but not the table itself)
 - may propagate to other tables if there are foreign key constraints with ON DELETE CASCADE

Data Modification

- update values in existing rows

```
UPDATE R  
SET <new value asgn>  
WHERE <condition>
```

 - <new value asgn> has the form attr = expr
 - separate multiple assignments with commas
 - expr can be an expression involving the same attribute or another – the value used is the one from the current row
 - all rows satisfying the condition are affected
 - only actual changes to the value are counted as updated rows
 - may propagate to other tables if there are foreign key constraints with ON UPDATE CASCADE

Data Modification

- any modification will be rejected if it violates foreign key, NOT NULL, and other integrity constraints

Quiz

PROJECT			
Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPARTMENT			
Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

Assuming that all default values are NULL, are these legal INSERT statements?

INSERT INTO PROJECT(Pname,Plocation,Dnum)
VALUES ("ProductA","Austin",1) 

INSERT INTO PROJECT(Pnumber,Pname,Plocation,Dnum)
VALUES (40,"ProductA","Austin",1) 

INSERT INTO PROJECT(Pnumber,Pname,Plocation,Dnum)
VALUES (40,"ProductA",NULL,1) 

INSERT INTO PROJECT(Pnumber,Pname,Plocation,Dnum)
VALUES (10,"ProductA","Austin",1) 

Quiz

PROJECT			
Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPARTMENT			
Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

Assuming that all default values are NULL, are these legal INSERT statements?

INSERT INTO PROJECT(Pnumber,Pname,Plocation,Dnum)
VALUES ("ProductA",40,"Austin",1) 

INSERT INTO PROJECT(Pnumber,Pname)
VALUES (40,"ProductA") 

Quiz

PROJECT			
Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPARTMENT			
Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS		
Dnumber	Dlocation	
1	Houston	
4	Stafford	
5	Bellaire	
5	Sugarland	
5	Houston	

Assuming that all default values are NULL, are these legal INSERT statements?

INSERT INTO PROJECT(Pnumber,Plocation,Dnum)
SELECT * FROM DEPT_LOCATIONS 

INSERT INTO PROJECT(Pnumber,Plocation,Dnum)
SELECT Dnumber+50,Dlocation,Dnumber FROM DEPT_LOCATIONS 

Quiz

How many rows will be deleted?

```
DELETE FROM WORKS_ON  
WHERE Hours < 10
```

```
DELETE FROM WORKS_ON W  
WHERE NOT EXISTS ( SELECT *  
                   FROM DEPENDENT D  
                   WHERE D.Essn=W.Essn )
```

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

Quiz

How many rows will be updated?

```
UPDATE WORKS_ON W  
SET Hours = NULL  
WHERE NOT EXISTS ( SELECT *  
                   FROM PROJECT P JOIN EMPLOYEE E  
                   ON P.Dnum=E.Dno  
                   WHERE W.Pno=P.Pnumber AND W.Essn=E.Ssn )
```

EXISTS is true if there are
rows in the subquery result
and false otherwise

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4