CSE 180: Lab Section 1

- Relational Algebra vs Relational Calculus
- Insert Operation
- Primary Key vs Unique
- Default
- NOT NULL
- IS NULL vs IS NOT NULL
- Expression in Select
- LIKE Operator
- Order By

Tuesday, October 17th
Utkarsh Gupta

Insert Operation

Syntax:

INSERT INTO TABLE_NAME (COL_1, COL_2, … , COL_N) VALUES (VAL_1, VAL_2, …, VAL_N);

```
INSERT INTO STUDENT (S_ID, S_NAME, SSN, DOB, GPA, HAS_GRAD, PHONE, EMAIL) VALUES (1, 'DAVID', '1234',
'1997-01-01', 3.8, false, '0123456', 'david@pqrs.com');

INSERT INTO STUDENT (S_ID, S_NAME, SSN, DOB, GPA, HAS_GRAD, PHONE, EMAIL) VALUES (2, 'JULIA', '4567',
'2000-02-18', 3.9, false, '1234589', 'julia@pqrs.com');

INSERT INTO STUDENT (S_ID, S_NAME, SSN, DOB, GPA, HAS_GRAD, PHONE, EMAIL) VALUES (3, 'DAVID', '2468',
'2000-02-01', 3.8, false, '9827123', 'david1@pqrs.com');
```

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com

Primary Key

Primary Key vs Unique

Primary Key

Primary key helps us to uniquely identify a row in the table.

The attribute **CANNOT** be **NULL**.

At most one Primary Key can be made for a given table.

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com

Primary Key vs Unique

```
CREATE TABLE STUDENT (
        S_ID INT,
        S_NAME VARCHAR(30),
        SSN CHAR(9),
        DOB DATE,
        GPA NUMERIC(3,2),
        HAS_GRAD BOOL,
        PHONE CHAR(10),
        EMAIL VARCHAR(50),
        PRIMARY KEY (S_ID)
);
```

Primary Key

Primary Key vs Unique

Unique

Unique

This also helps up to identify a unique row in the table.

The attribute **CAN** be **NULL**.

NULL value is used when you don't know the value of an attribute of a given tuple.

There can be multiple unique constraints along with a primary key in the table.

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	**1234**	1997-01-01	3.8	**F**	**0123456**	**david@pqrs.com**
2	JULIA	**4567**	2000-02-18	3.9	**F**	**1234589**	**julia@pqrs.com**
3	DAVID	**2468**	2000-02-01	3.8	**F**	**9827123**	**david1@pqrs.com**
4	JOEL	**3412**	2002-03-23	3.34	**F**	**NULL**	**joel@pqrs.com**
5	ABY	**4321**	**NULL**	3.32	**T**	**NULL**	**NULL**

Primary Key vs Unique

```
CREATE TABLE STUDENT (
        S_ID INT,
        S_NAME VARCHAR(30),
        SSN CHAR(9),
        DOB DATE,
        GPA NUMERIC(3,2),
        HAS_GRAD BOOL,
        PHONE CHAR(10),
        EMAIL VARCHAR(50),
        PRIMARY KEY (S_ID),
        UNIQUE (EMAIL),
        UNIQUE (PHONE),
        UNIQUE (SSN)
);
```

Primary Key

Primary Key vs Unique

Unique

Unique

This also helps up to identify a unique row in the table.

The attribute **CAN** be **NULL**.

There can be multiple unique constraints along with a primary key in the table.

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	**0123456**	**david@pqrs.com**
2	JULIA	4567	2000-02-18	3.9	F	**1234589**	**julia@pqrs.com**
3	DAVID	2468	2000-02-01	3.8	F	**9827123**	**david1@pqrs.com**
4	JOEL	3412	2002-03-23	3.34	F	**NULL**	**joel@pqrs.com**
5	ABY	4321	**NULL**	3.32	T	**NULL**	**NULL**

```
INSERT INTO STUDENT VALUES (4, 'JOEL', '3412', '2002-03-23', 3.34, false, NULL, 'joel@pqrs.com');
INSERT INTO STUDENT VALUES (5, 'ABY', '4321', NULL, 3.32, true, NULL, NULL);
```

Primary Key vs Unique

```
CREATE TABLE STUDENT (
        S_ID INT,
        S_NAME VARCHAR(30),
        SSN CHAR(9),
        DOB DATE,
        GPA NUMERIC(3,2),
        HAS_GRAD BOOL,
        PHONE CHAR(10),
        EMAIL VARCHAR(50),
        PRIMARY KEY (S_ID),
        UNIQUE (EMAIL),
        UNIQUE (PHONE, S_NAME),
        UNIQUE (SSN)
);
```

Default

Default

Specifies a Default value for the attribute in the case when the respective value is missing.

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com
4	JOEL	3412	2002-03-23	3.34	F	NULL	joel@pqrs.com
5	ABY	4321	NULL	3.32	T	NULL	NULL

Default (cont.)

```
CREATE TABLE STUDENT (
        S_ID INT,
        S_NAME VARCHAR(30),
        SSN CHAR(9),
        DOB DATE DEFAULT '2000-01-01',
        GPA NUMERIC(3,2),
        HAS_GRAD BOOL,
        PHONE CHAR(10),
        EMAIL VARCHAR(50),
        PRIMARY KEY (S_ID),
        UNIQUE (EMAIL),
        UNIQUE (PHONE, S_NAME),
        UNIQUE (SSN)
);
```

Default (cont.)

Default

Specifies a Default value for the attribute in the case when the respective value is missing.

```
INSERT INTO STUDENT (S_ID, S_NAME, SSN, GPA, HAS_GRAD, PHONE,
EMAIL) VALUES (5, 'ABY', '4321', 3.32, false, NULL, NULL);
```

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com
4	JOEL	3412	2002-03-23	3.34	F	NULL	joel@pqrs.com
5	ABY	4321	**2000-01-01**	3.32	F	NULL	NULL

Default (cont.)

Default

Specifies a Default value for the attribute in the case when the respective value is missing.

```
INSERT INTO STUDENT VALUES (5, 'ABY', '4321', NULL, 3.32, true, NULL, NULL);
```

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com
4	JOEL	3412	2002-03-23	3.34	F	NULL	joel@pqrs.com
5	ABY	4321	2000-01-01 / NULL ?	3.32	F	NULL	NULL

Default (cont.)

Default

Specifies a Default value for the attribute in the case when the respective value is missing.

```
INSERT INTO STUDENT VALUES (5, 'ABY', '4321', NULL, 3.32, true, NULL, NULL);
```

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com
4	JOEL	3412	2002-03-23	3.34	F	NULL	joel@pqrs.com
5	ABY	4321	**NULL**	3.32	F	NULL	NULL

Correct Answer

NOT NULL

NOT NULL

NOT NULL property will help you to specify that an attribute **CANNOT** have NULL values.

Previously we had:

```
INSERT INTO STUDENT VALUES (5, 'ABY', '4321', NULL, 3.32, true, NULL, NULL);
```

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com
4	JOEL	3412	2002-03-23	3.34	F	NULL	joel@pqrs.com
5	ABY	4321	**NULL**	3.32	F	NULL	NULL

Default (cont.)

```
CREATE TABLE STUDENT (
        S_ID INT,
        S_NAME VARCHAR(30),
        SSN CHAR(9),
        DOB DATE DEFAULT '2000-01-01' NOT NULL,
        GPA NUMERIC(3,2),
        HAS_GRAD BOOL,
        PHONE CHAR(10),
        EMAIL VARCHAR(50),
        PRIMARY KEY (S_ID),
        UNIQUE (EMAIL),
        UNIQUE (PHONE, S_NAME),
        UNIQUE (SSN)
);
```

NOT NULL

NOT NULL

NOT NULL property will help you to specify that an attribute **CANNOT** have NULL values.

The following will FAIL:

```
INSERT INTO STUDENT VALUES (5, 'ABY', '4321', NULL, 3.32, true, NULL, NULL);
```

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com
4	JOEL	3412	2002-03-23	3.34	F	NULL	joel@pqrs.com
5	ABY	4321	**NULL**	3.32	F	NULL	NULL

NOT NULL

NOT NULL

NOT NULL property will help you to specify that an attribute **CANNOT** have NULL values.

```
INSERT INTO STUDENT (S_ID, S_NAME, SSN, GPA, HAS_GRAD, PHONE, EMAIL) VALUES (5,
'ABY', '4321', 3.32, true, NULL, NULL);
```

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com
4	JOEL	3412	2002-03-23	3.34	F	NULL	joel@pqrs.com
5	ABY	4321	**2000-01-01**	3.32	F	NULL	NULL

IS NULL and IS NOT NULL

List of students whose GPA is greater than 3.33 AND Phone is not NULL

List of students whose GPA is greater than 3.33 AND Phone is NULL

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com
4	JOEL	3412	2002-03-23	3.34	F	NULL	joel@pqrs.com
5	ABY	4321	2000-01-01	3.32	F	NULL	NULL

What are the outputs of the following queries?

```
SELECT * FROM STUDENT
WHERE GPA > 3.33 AND PHONE != NULL;
```

```
SELECT * FROM STUDENT
WHERE GPA > 3.33 AND PHONE = NULL;
```

What's wrong?

IS NULL and IS NOT NULL (cont.)

List of students whose GPA is greater than 3.33 AND Phone is not NULL

List of students whose GPA is greater than 3.33 AND Phone is NULL

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com
4	JOEL	3412	2002-03-23	3.34	F	NULL	joel@pqrs.com
5	ABY	4321	2000-01-01	3.32	F	NULL	NULL

What are the outputs of the following queries?

```
SELECT * FROM STUDENT
WHERE GPA > 3.33 AND PHONE IS NOT NULL;
```

```
SELECT * FROM STUDENT
WHERE GPA > 3.33 AND PHONE IS NULL;
```

IS NULL and IS NOT NULL (cont.)

List of students whose GPA is greater than 3.33 **OR** Phone is not NULL

List of students whose GPA is greater than 3.33 **OR** Phone is NULL

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com
4	JOEL	3412	2002-03-23	3.34	F	NULL	joel@pqrs.com
5	ABY	4321	2000-01-01	3.32	F	NULL	NULL

What is the output of the following query?

```
SELECT * FROM STUDENT
WHERE GPA > 3.33 OR PHONE != NULL;
```

```
SELECT * FROM STUDENT
WHERE GPA > 3.33 OR PHONE = NULL;
```

What do you think? Is something wrong here?

IS NULL and IS NOT NULL (cont.)

List of students whose GPA is greater than 3.33 **OR** Phone is not NULL

List of students whose GPA is greater than 3.33 **OR** Phone is NULL

S_ID	S_NAME	SSN	DOB	GPA	HAS_GRAD	PHONE	Email
1	DAVID	1234	1997-01-01	3.8	F	0123456	david@pqrs.com
2	JULIA	4567	2000-02-18	3.9	F	1234589	julia@pqrs.com
3	DAVID	2468	2000-02-01	3.8	F	9827123	david1@pqrs.com
4	JOEL	3412	2002-03-23	3.34	F	NULL	joel@pqrs.com
5	ABY	4321	2000-01-01	3.32	F	NULL	NULL

What is the output of the following query?

```
SELECT * FROM STUDENT
WHERE GPA > 3.33 OR PHONE IS NOT NULL;
```

```
SELECT * FROM STUDENT
WHERE GPA > 3.33 OR PHONE IS NULL;
```

Courses table

```
CREATE TABLE COURSES (
        C_ID INT PRIMARY KEY,
        C_NAME CHAR(9) UNIQUE,
        CREDITS INT DEFAULT 5 NOT NULL
);
```

```
INSERT INTO COURSES VALUES (180, 'DB 1');
INSERT INTO COURSES VALUES (181, 'DB 2');
INSERT INTO COURSES VALUES (215, 'DB GRAD 1');
INSERT INTO COURSES VALUES (160, 'CG');
INSERT INTO COURSES VALUES (280, 'CV SEM', 2);
```

Enrolment table

```
CREATE TABLE ENROLMENT (
        S_ID INT REFERENCES STUDENT,
        C_ID INT REFERENCES COURSES,
        DROPPED BOOL NOT NULL,
        PRIMARY KEY(S_ID, C_ID)
);
```

```
INSERT INTO ENROLMENT VALUES (1, 180, false);
INSERT INTO ENROLMENT VALUES (5, 160, false);
INSERT INTO ENROLMENT VALUES (5, 215, false);
INSERT INTO ENROLMENT VALUES (1, 181, true);
INSERT INTO ENROLMENT VALUES (2, 160, false);
INSERT INTO ENROLMENT VALUES (5, 180, false);
INSERT INTO ENROLMENT VALUES (2, 280, false);
INSERT INTO ENROLMENT VALUES (5, 181, false);
INSERT INTO ENROLMENT VALUES (2, 181, true);
INSERT INTO ENROLMENT VALUES (3, 180, false);
INSERT INTO ENROLMENT VALUES (5, 280, false);
```

DISTINCT

List the DISTINCT student ids who are enrolled in any course.

```
SELECT S_ID FROM ENROLMENT;
SELECT DISTINCT S_ID FROM ENROLMENT;
```

List the DISTINCT student names who are enrolled in any course.

```
SELECT S_NAME FROM STUDENT S, ENROLMENT E
WHERE S.S_ID = E.S_ID;

SELECT DISTINCT S_NAME FROM STUDENT S, ENROLMENT E
WHERE S.S_ID = E.S_ID;
```

Expressions in SELECT

Display the scaled GPA to a range of 0-10;

SELECT S_NAME, **(GPA * 10)/4** AS SCALED_GPA FROM STUDENT;

Get a better result:

SELECT S_NAME, **ROUND((GPA * 10)/4, 2)** AS SCALED_GPA FROM STUDENT;

LIKE Operator

INSERT THE VALUES:

```
INSERT INTO STUDENT VALUES (6, 'UTKARSH''S', '1235', '1997-01-01', 3.8, false, '0132346', 'utk@pqrs.com');
INSERT INTO STUDENT VALUES (7, 'ST''MIDDLE''EN', '8412', '2000-02-18', 3.9, false, '1891231', 'random@pqrs.com');
INSERT INTO STUDENT VALUES (8, 'CATHY''''S', '8274', '2000-09-23', 3.8, true, '817423', 'cathy@pqrs.com');
```

```
 s_id |    s_name    |    ssn    |    dob     | gpa  | has_grad |   phone    |      email
------+--------------+-----------+------------+------+----------+------------+-----------------
    1 | DAVID        | 1234      | 1997-01-01 | 3.80 | f        | 0123456    | david@pqrs.com
    2 | JULIA        | 4567      | 2000-02-18 | 3.90 | f        | 1234589    | julia@pqrs.com
    3 | DAVID        | 2468      | 2000-02-01 | 3.80 | f        | 9827123    | david1@pqrs.com
    4 | JOEL         | 3412      | 2002-03-23 | 3.34 | f        |            | joel@pqrs.com
    5 | ABY          | 4321      | 2000-01-01 | 3.32 | t        |            |
    6 | UTKARSH'S    | 1235      | 1997-01-01 | 3.80 | f        | 0132346    | utk@pqrs.com
    7 | ST'MIDDLE'EN | 8412      | 2000-02-18 | 3.90 | f        | 1891231    | random@pqrs.com
    8 | CATHY''S     | 8274      | 2000-09-23 | 3.80 | t        | 817423     | cathy@pqrs.com
(8 rows)
```

LIKE Operator

Find the S_Name where it either starts from a letter ‘S’ or ‘U’

```
SELECT * FROM STUDENT WHERE S_NAME LIKE 'S%' OR S_NAME LIKE 'U%';
```

```
 s_id |    s_name    |    ssn    |    dob     | gpa  | has_grad |   phone    |      email
------+--------------+-----------+------------+------+----------+------------+-----------------
    6 | UTKARSH'S    | 1235      | 1997-01-01 | 3.80 | f        | 0132346    | utk@pqrs.com
    7 | ST'MIDDLE'EN | 8412      | 2000-02-18 | 3.90 | f        | 1891231    | random@pqrs.com
(2 rows)
```

LIKE Operator

Find the tuples where S_Name contains a single quote

SELECT * FROM STUDENT WHERE S_NAME LIKE '%''%';

```
 s_id |    s_name    |    ssn    |    dob     | gpa  | has_grad |   phone    |       email
------+--------------+-----------+------------+------+----------+------------+-------------------
    6 | UTKARSH'S    | 1235      | 1997-01-01 | 3.80 | f        | 0132346    | utk@pqrs.com
    7 | ST'MIDDLE'EN | 8412      | 2000-02-18 | 3.90 | f        | 1891231    | random@pqrs.com
    8 | CATHY''S     | 8274      | 2000-09-23 | 3.80 | t        | 817423     | cathy@pqrs.com
(3 rows)
```

LIKE Operator

Find the tuples where S_Name contains at least two single quotes

SELECT * FROM STUDENT WHERE S_NAME LIKE '%''%''%';

```
 s_id |    s_name     |    ssn     |    dob     | gpa  | has_grad |   phone    |       email
------+---------------+------------+------------+------+----------+------------+-------------------
    7 | ST'MIDDLE'EN | 8412       | 2000-02-18 | 3.90 | f        | 1891231    | random@pqrs.com
    8 | CATHY''S     | 8274       | 2000-09-23 | 3.80 | t        | 817423     | cathy@pqrs.com
(2 rows)
```

LIKE Operator

Find the output of the following:

SELECT * FROM STUDENT WHERE S_NAME LIKE '%''%''_';

```
 s_id |  s_name  |    ssn    |    dob     | gpa  | has_grad |   phone    |     email
------+----------+-----------+------------+------+----------+------------+---------------
    8 | CATHY''S | 8274      | 2000-09-23 | 3.80 | t        | 817423     | cathy@pqrs.com
(1 row)
```

SELECT * FROM STUDENT WHERE S_NAME LIKE '%''''_';

```
 s_id |  s_name  |    ssn    |    dob     | gpa  | has_grad |   phone    |     email
------+----------+-----------+------------+------+----------+------------+---------------
    8 | CATHY''S | 8274      | 2000-09-23 | 3.80 | t        | 817423     | cathy@pqrs.com
(1 row)
```

LIKE Operator

Find the output of the following:

SELECT * FROM STUDENT WHERE S_NAME LIKE '_____''_';

```
 s_id |  s_name  |    ssn    |    dob     | gpa  | has_grad |   phone    |     email
------+----------+-----------+------------+------+----------+------------+---------------
    8 | CATHY''S | 8274      | 2000-09-23 | 3.80 | t        | 817423     | cathy@pqrs.com
(1 row)
```

ORDER BY

Order by displays the result in sorted order.

SELECT [DISTINCT] <list of attributes>
FROM $R_1, R_2, \ldots, R_n$
[WHERE *condition*]
[ORDER BY <list of attributes>]

Reference: Lecture 3 slides

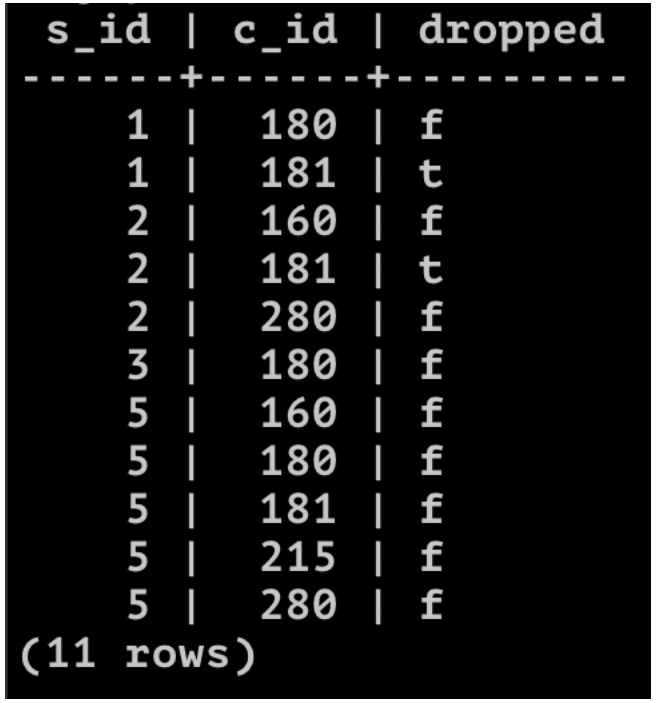
ORDER BY

SELECT * FROM ENROLMENT;

```
 s_id | c_id | dropped
------+------+---------
    1 |  180 | f
    5 |  160 | f
    5 |  215 | f
    1 |  181 | t
    2 |  160 | f
    5 |  180 | f
    2 |  280 | f
    5 |  181 | f
    2 |  181 | t
    3 |  180 | f
    5 |  280 | f
(11 rows)
```

ORDER BY

SELECT * FROM ENROLMENT ORDER BY S_ID;



s_id	c_id	dropped
1	180	f
1	181	t
2	160	f
2	181	t
2	280	f
3	180	f
5	160	f
5	180	f
5	181	f
5	215	f
5	280	f

(11 rows)

ORDER BY

SELECT * FROM ENROLMENT ORDER BY S_ID, C_ID DESC;

```
 s_id | c_id | dropped
------+------+---------
    1 |  181 | t
    1 |  180 | f
    2 |  280 | f
    2 |  181 | t
    2 |  160 | f
    3 |  180 | f
    5 |  280 | f
    5 |  215 | f
    5 |  181 | f
    5 |  180 | f
    5 |  160 | f
(11 rows)
```

Insert Commands

General

```
INSERT INTO STUDENT VALUES (1, 'DAVID', '1234', '1997-01-01', 3.8, false, '0123456', 'david@pqrs.com');
INSERT INTO STUDENT VALUES (2, 'JULIA', '4567', '2000-02-18', 3.9, false, '1234589', 'julia@pqrs.com');
INSERT INTO STUDENT VALUES (3, 'DAVID', '2468', '2000-02-01', 3.8, false, '9827123', 'david1@pqrs.com');
INSERT INTO STUDENT VALUES (4, 'JOEL', '3412', '2002-03-23', 3.34, false, NULL, 'joel@pqrs.com');
INSERT INTO STUDENT (S_ID, S_NAME, SSN, GPA, HAS_GRAD, PHONE, EMAIL) VALUES (5, 'ABY', '4321', 3.32, false, NULL, NULL);
```

Unique

```
INSERT INTO STUDENT VALUES (5, ‘ABY', ‘4321', NULL, 3.32, true, NULL, NULL);
```

Default

```
INSERT INTO STUDENT (S_ID, S_NAME, SSN, GPA, HAS_GRAD, PHONE, EMAIL) VALUES (5, 'ABY', '4321', 3.32, true, NULL, NULL);
```

Default With NULL

```
INSERT INTO STUDENT VALUES (5, 'ABY', '4321', NULL, 3.32, true, NULL, NULL);
```

Insert Commands

LIKE OPERATOR

```
INSERT INTO STUDENT VALUES (6, 'UTKARSH''S', '1235', '1997-01-01', 3.8, false, '0132346', 'utk@pqrs.com');
INSERT INTO STUDENT VALUES (7, 'ST''MIDDLE''EN', '8412', '2000-02-18', 3.9, false, '1891231', 'random@pqrs.com’);
INSERT INTO STUDENT VALUES (8, 'CATHY''''S', '8274', '2000-09-23', 3.8, true, '817423', 'cathy@pqrs.com');
```