COMPS320F Database Management

2021.10.4 Jiawei Wang

Lecture 3

SQL: DATA MANIPULATION PART I

Terri Wong

School of Science and Technology

Content

- Purpose and importance of SQL.
- How to retrieve data from database using SELECT and:
  - Use compound WHERE conditions.
  - Sort query results using ORDER BY.
  - Use aggregate functions.
  - Group data using GROUP BY and HAVING.
  - Use subqueries.

Content

- Join tables together.
- Perform set operations (UNION, INTERSECT, EXCEPT).

- How to update database using INSERT, UPDATE, and DELETE.

Database Language

- Ideally, database language should allow user to:
  - create the database and relation structures;
  - perform insertion, modification, deletion of data from relations;
  - perform simple and complex queries.
- Must perform these tasks with minimal user effort and command structure/syntax must be easy to learn.
- It must be portable.

SQL

- SQL is a transform-oriented language with 2 major components:
  - A DML for retrieving and updating data. Manipulating Language (操控)
  - A DDL for defining database structure.

Create Table Statement
Create Table / View
Alter Table / view

SQL

- Consists of standard English words:

1) CREATE TABLE Staff(staffNo VARCHAR(5),
lName VARCHAR(15),
salary DECIMAL(7,2));

2) INSERT INTO Staff VALUES ('SG16', 'Brown', 8300);

Case Insensitive

3) SELECT staffNo, lName, salary
FROM Staff
WHERE salary > 10000;

Writing SQL Commands

- Most components of an SQL statement are ***case insensitive***, except for literal character data.
- More readable with indentation and lineation:
  - Each clause should begin on a new line.
  - Start of a clause should line up with start of other clauses.
  - If clause has several parts, should each appear on a separate line and be indented under start of clause.

Literals

- Literals are constants used in SQL statements.
- All non-numeric literals must be enclosed in ***single quotes***

  e.g. ‘L’, or ‘London’, etc.
- All numeric literals must not be enclosed in quotes

  e.g. 6, or 650, or 650.5, etc.

Sample Schema: DreamHome (property agent)

Branch

BRANCHNO	STREET	CITY	POSTCODE
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

http://learn.ouhk.edu.hk/~t810870/320f/lecture/

Staff

STAFFNO	FNAME	LNAME	POSITION	SEX	DOB	SALARY	BRANCHNO
SL21	John	White	Manager	M	01-OCT-99	30000	B005
SG37	Ann	Beech	Assistant	F	10-NOV-97	12000	B003
SG14	David	Ford	Supervisor	M	24-MAR-00	18000	B003
SA9	Mary	Howe	Assistant	F	19-FEB-01	9000	B007
SG5	Susan	Brand	Manager	F	03-JUN-90	24000	B003
SL41	Julie	Lee	Assistant	F	13-JUN-92	9000	B005

PropertyForRent

PROPERTYNO	STREET	CITY	POSTCODE	TYPE	ROOMS	RENT	OWNERNO	STAFFNO	BRANCHNO
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40	(null)	B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

Client

CLIENTNO	FNAME	LNAME	TELNO	PREFTYPE	MAXRENT	EMAIL
CR76	John	Kay	0171-774-5632	Flat	425	john.kay@gmail.com
CR56	Aline	Steward	0141-848-1825	Flat	350	astewart@hotmail.com
CR74	Mike	Ritchie	01475-943-1728	House	750	mritchie@yahoo.co.uk
CR62	Mary	Tregear	01224-196720	Flat	600	maryt@hotmail.co.uk

PrivateOwner

OWNERNO	FNAME	LNAME	ADDRESS	TELNO	EMAIL
CO46	Joe	Keogh	2 Fergus Dr. Aberdeen AB2	01224-861212	jkeogh@lhh.com
CO87	Carol	Farrel	6 Achray St. Glasgow G32 9DX	0141-357-7419	cfarrel@gmail.com
CO40	Tina	Murphy	63 Well St. Glasgow G42	0141-943-1728	tinam@hotmail.com
CO93	Tony	Shaw	12 Park Pl. Glasgow G4 0QR	0141-225-7025	tony.shaw@ark.com

Viewing

CLIENTNO	PROPERTYNO	VIEWDATE	REMARK
CR56	PA14	24-MAY-19	too small
CR76	PG4	20-APR-19	too remote
CR56	PG4	26-MAY-19	(null)
CR62	PA14	14-MAY-19	no dining room
CR56	PG36	28-APR-19	(null)

Registration

CLIENTNO	BRANCHNO	STAFFNO	DATEJOINED
CR76	B005	SL41	13-JAN-19
CR56	B003	SG37	13-APR-19
CR74	B003	SG37	16-NOV-19
CR62	B007	SA9	07-MAR-19

SELECT Statement

FROM Must Exist	Specifies table(s) to be used.
WHERE	Filters rows.
GROUP BY	Forms groups of rows with same column value.
HAVING	Filters groups subject to some condition.
SELECT	Specifies which columns are to appear in output.
ORDER BY	Specifies the order of the output.

optional

Example 1 All Columns, All Rows

List full details of all staff.

```
SELECT staffNo, fName, lName, address,
       position, sex, DOB, salary, branchNo
FROM Staff;
```

Which columns to be selected

Which Table to be selected

Can use ***** as an abbreviation for '**all columns**':

```
SELECT *
FROM Staff;
```

Example 1 All Columns, All Rows

Table 5.1 Result table for Example 5.1.

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005

Example 2 Specific Columns, All Rows

- Produce a list of salaries for all staff, showing only staff number, first and last names, and salary.

```
SELECT staffNo, fName, lName, salary
FROM Staff;
```

Example 2 Specific Columns, All Rows

Table 5.2 Result table for Example 5.2.

staffNo	fName	lName	salary
SL21	John	White	30000.00
SG37	Ann	Beech	12000.00
SG14	David	Ford	18000.00
SA9	Mary	Howe	9000.00
SG5	Susan	Brand	24000.00
SL41	Julie	Lee	9000.00

Example 3 Use of DISTINCT

- List the property numbers of all properties that have been viewed.

```
SELECT propertyNo
FROM Viewing;
```

propertyNo
PA14
PG4
PG4
PA14
PG36

Example 3 Use of DISTINCT

- Use DISTINCT to eliminate duplicates:

```
SELECT DISTINCT propertyNo
FROM Viewing;
```

propertyNo
PA14
PG4
PG36

Note no duplicate values of 'PA14' and 'PG4' in the result set

Example 4 Calculated Fields

- Produce list of monthly salaries for all staff, showing staff number, first/last name, and salary.

```
SELECT staffNo, fName, lName, salary/12
FROM Staff;
```

staffno character(5)	fname character varying(10)	lname character varying(10)	?column? integer
SL21	John	White	2500
SG37	Ann	Beech	1000
SG14	David	Ford	1500
SA9	Mary	Howe	750
SG5	Susan	Brand	2000
SL41	Julie	Lee	750

Example 4 Calculated Fields

- To name column, use AS clause:

```
SELECT staffNo, fName, lName,
       salary/12 AS monthlySalary
FROM Staff;
```

staffno character(5)	fname character varying(10)	lname character varying(10)	monthlysalary integer
SL21	John	White	2500
SG37	Ann	Beech	1000
SG14	David	Ford	1500
SA9	Mary	Howe	750
SG5	Susan	Brand	2000
SL41	Julie	Lee	750

Example 5 Comparison Search Condition

- List all staff with a salary greater than 10,000.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary > 10000;
```

Table 5.5 Result table for Example 5.5.

staffNo	fName	lName	position	salary
SL21	John	White	Manager	30000.00
SG37	Ann	Beech	Assistant	12000.00
SG14	David	Ford	Supervisor	18000.00
SG5	Susan	Brand	Manager	24000.00

Example 6.1 Compound Comparison Search Condition

- List addresses of all branch offices in London or Glasgow.

```
SELECT *
FROM Branch
WHERE city = 'London' OR city = 'Glasgow';
```

branchno character(5)	street character varying(35)	city character varying(10)	postcode character varying(10)
B005	22 Deer Rd	London	SW1 4EH
B003	163 Main St	Glasgow	G11 9QX
B002	56 Clover Dr	London	NW10 6EU

Example 6.2 Compound Comparison Search Condition

- List all staff with a salary greater less than 20,000 or greater than 30,000.

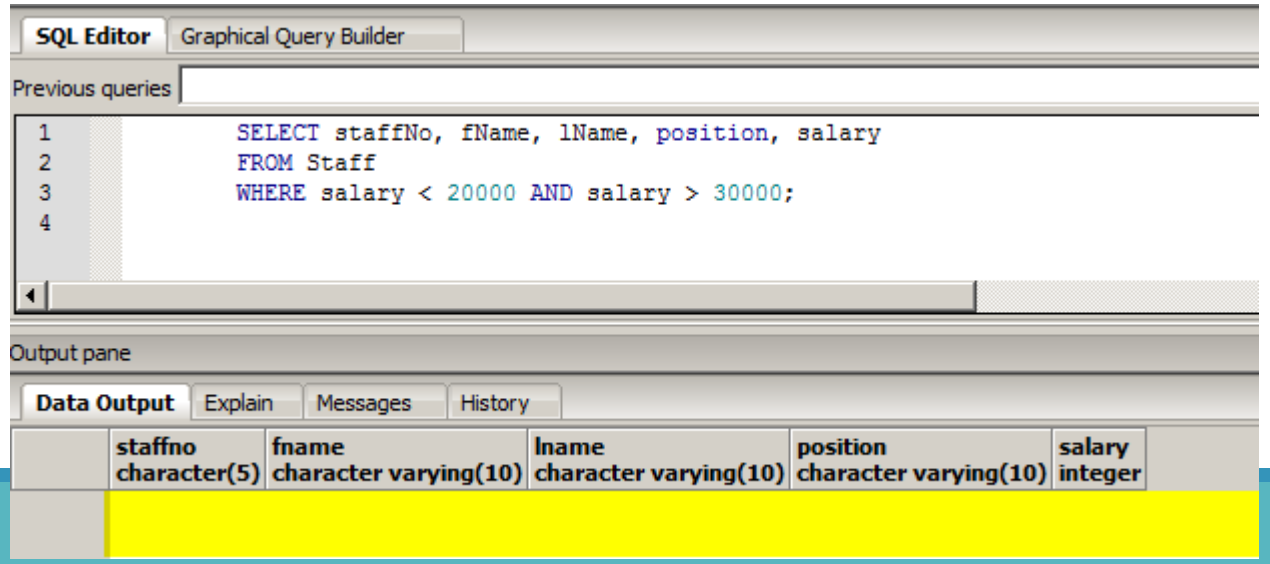
```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary < 20000 OR salary > 30000;
```

staffno character(5)	fname character varying(10)	lname character varying(10)	position character varying(10)	salary integer
SG37	Ann	Beech	Assistant	12000
SG14	David	Ford	Supervisor	18000
SA9	Mary	Howe	Assistant	9000
SL41	Julie	Lee	Assistant	9000

Question

- Why the following query returns no result set?

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary < 20000 AND salary > 30000;
```



Example 7.1 Range Search Condition

- List all staff with a salary between 20,000 and 30,000.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary BETWEEN 20000 AND 30000;
```

  - *Note: BETWEEN test includes the endpoints of range.*

Example 7 Range Search Condition

Table 5.7 Result table for Example 5.7.

staffNo	fName	lName	position	salary
SL21	John	White	Manager	30000.00
SG5	Susan	Brand	Manager	24000.00

Example 7.2 Range Search Condition

- Also a negated version NOT BETWEEN.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary>=20000 AND salary <= 30000;
```

It is possible not selecting colume salary in select statement

=> The reason to include it is to provide a better way for checking in output

Example 8 Set Membership

- List all managers and supervisors.

```
SELECT staffNo, fName, lName, position
FROM Staff
WHERE position IN ('Manager', 'Supervisor');
```

Table 5.8 Result table for Example 5.8.

staffNo	fName	lName	position
SL21	John	White	Manager
SG14	David	Ford	Supervisor
SG5	Susan	Brand	Manager

Example 8 Set Membership

- There is a negated version (NOT IN).
- IN does not add much to SQL's expressive power. Could have expressed this as:

```
SELECT staffNo, fName, lName, position
    FROM Staff
WHERE position='Manager' OR
        position='Supervisor';
```

Note: IN is more efficient when set contains many values.

Example 9 Pattern Matching

- SQL has two special pattern matching symbols:
  - %: sequence of ***zero or more*** characters;
  - _ (underscore): any ***single character.***
- LIKE ‘%Glasgow%’ means a sequence of characters of any length containing ‘Glasgow’.

Example 9.1 Pattern Matching

- Find all owners with the string ‘Glasgow’ in their address.

```
SELECT ownerNo, fName, lName, address, telNo
FROM PrivateOwner
WHERE address LIKE ‘%Glasgow%’;
```

Table 5.9 Result table for Example 5.9.

ownerNo	fName	lName	address	telNo
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX	0141-357-7419
CO40	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728
CO93	Tony	Shaw	12 Park Pl, Glasgow G4 0QR	0141-225-7025

Example 9.2 Pattern Matching

- Find all staff having the character ‘n’ as the last character in their first names.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE fname like '%n';
```

staffno character(5)	fname character varying(10)	lname character varying(10)	position character varying(10)	salary integer
SL21	John	White	Manager	30000
SG37	Ann	Beech	Assistant	12000
SG5	Susan	Brand	Manager	24000

Example 9.3 Pattern Matching

- Find all staff having one single character ending with the string ‘nn’ as the last two characters in their first names.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE fname like ‘_nn';
```

staffno character(5)	fname character varying(10)	lname character varying(10)	position character varying(10)	salary integer
SG37	Ann	Beech	Assistant	12000

Question

Show clientNo, fname, email of clients whose email addresses are from Hotmail accounts.

SELECT

FROM

WHERE

CLIENTNO	FNAME	EMAIL
CR56	Aline	astewart@hotmail.com
CR62	Mary	maryt@hotmail.co.uk

Answer

Show clientNo, fname, email of clients whose email addresses are from Hotmail accounts.

```
select clientNo, fname, email
from client
where email like '%@hotmail%';
```

CLIENTNO	FNAME	EMAIL
CR56	Aline	astewart@hotmail.com
CR62	Mary	maryt@hotmail.co.uk

Question

Show clientNo, fname, email of clients whose email addresses are from Hotmail accounts ***in UK only.***

SELECT

FROM

WHERE

CLIENTNO	FNAME	EMAIL
CR62	Mary	maryt@hotmail.co.uk

Answer

Show clientNo, fname, email of clients whose email addresses are from Hotmail accounts ***in UK only.***

```
select clientNo, fname, email
from client
where email like '%@hotmail%uk';
```

CLIENTNO	FNAME	EMAIL
CR62	Mary	maryt@hotmail.co.uk

Example 10 NULL Search Condition

- List details of all viewings on property PG4 where a comment has not been supplied.
- There are 2 viewings for property PG4, one with and one without a comment.

```
SELECT clientNo, viewDate, remark
FROM Viewing
WHERE propertyNo = 'PG4' ;
```

clientno character(5)	viewdate date	remark character varying(15)
CR76	2019-04-20	too remote
CR56	2019-05-26	

- Have to test for null explicitly using special keyword IS NULL:

```
SELECT clientNo, viewDate, remark
FROM Viewing
WHERE propertyNo = 'PG4'
AND remark IS NULL;
```

clientno character(5)	viewdate date	remark character varying(15)
CR56	2019-05-26	

Example 10 NULL Search Condition

Negated version (IS NOT NULL) can test for non-null values.

```
SELECT clientNo, viewDate, remark
FROM Viewing
WHERE propertyNo = 'PG4'
AND remark IS NOT NULL;
```

clientno character(5)	viewdate date	remark character varying(15)
CR76	2019-04-20	too remote

Example 11 Single Column Ordering

- List salaries for all staff, arranged in ***ascending*** order of salary.

```
SELECT staffNo, fName, lName, salary
FROM Staff
ORDER BY salary;
```

staffno character(5)	fname character varying(10)	lname character varying(10)	salary integer
SA9	Mary	Howe	9000
SL41	Julie	Lee	9000
SG37	Ann	Beech	12000
SG14	David	Ford	18000
SG5	Susan	Brand	24000
SL21	John	White	30000

Example 11 Single Column Ordering

- List salaries for all staff, arranged in ***descending*** order of salary.

```
SELECT staffNo, fName, lName, salary
FROM Staff
ORDER BY salary desc;
```

staffno character(5)	fname character varying(10)	lname character varying(10)	salary integer
SL21	John	White	30000
SG5	Susan	Brand	24000
SG14	David	Ford	18000
SG37	Ann	Beech	12000
SA9	Mary	Howe	9000
SL41	Julie	Lee	9000

Example 12 Multiple Column Ordering

- Produce abbreviated list of properties in order of property type.

```
SELECT propertyNo, type, rooms, rent
FROM PropertyForRent
ORDER BY type;
```

Example 12 Multiple Column Ordering

Table 5.12(a) Result table for Example 5.12 with one sort key.

propertyNo	type	rooms	rent
PL94	Flat	4	400
PG4	Flat	3	350
PG36	Flat	3	375
PG16	Flat	4	450
PA14	House	6	650
PG21	House	5	600

Example 12 Multiple Column Ordering

- Four flats in this list - as no minor sort key specified, system arranges these rows in any order it chooses.
- To arrange in order of rent, specify minor order:

```
SELECT propertyNo, type, rooms, rent
FROM PropertyForRent
ORDER BY type, rent DESC;
```

Example 12 Multiple Column Ordering

Table 5.12(b) Result table for Example 5.12 with two sort keys.

propertyNo	type	rooms	rent
PG16	Flat	4	450
PL94	Flat	4	400
PG36	Flat	3	375
PG4	Flat	3	350
PA14	House	6	650
PG21	House	5	600

SELECT Statement - Aggregates

- ISO standard defines five aggregate functions:

Output: One Line

COUNT returns number of values in specified column.

SUM returns sum of values in specified column.

AVG returns average of values in specified column.

MIN returns smallest value in specified column.

MAX returns largest value in specified column.

SELECT Statement - Aggregates

- Each operates on a single column of a table and returns a single value.
- COUNT, MIN, and MAX apply to numeric and non-numeric fields, but SUM and AVG may be used on numeric fields only.
- Apart from COUNT(*), each function eliminates nulls first and operates only on remaining non-null values.

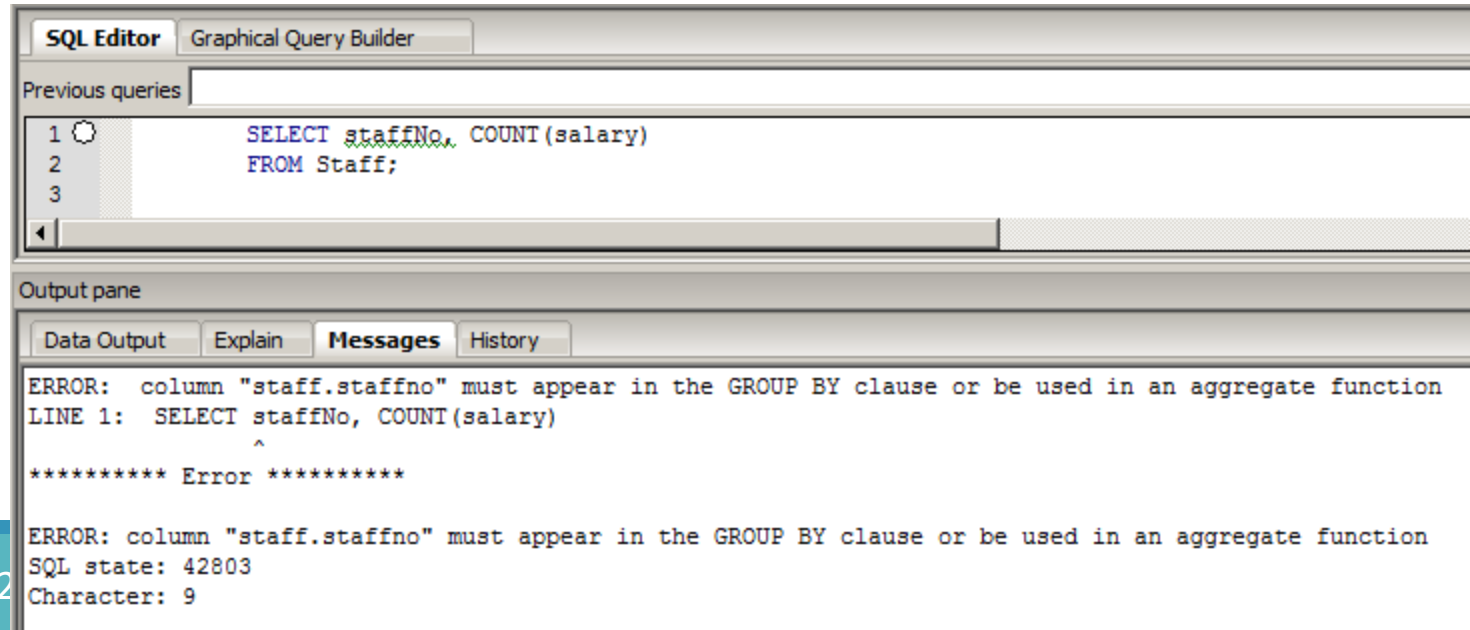
SELECT Statement - Aggregates

- COUNT(*) counts all rows of a table, regardless of whether nulls or duplicate values occur.
- Can use DISTINCT before column name to eliminate duplicates.
- DISTINCT has no effect with MIN/MAX, but may have with SUM/AVG.

SELECT Statement - Aggregates

- Aggregate functions can be used only in SELECT list and in HAVING clause.
- If SELECT list includes an aggregate function and there is no GROUP BY clause, SELECT list cannot reference a column without an aggregate function. For example, the following is illegal:

```
SELECT staffNo, COUNT(salary)
FROM Staff;
```



Question

- What is the outputs of the following statements:

```
select count(*) from viewing;
```

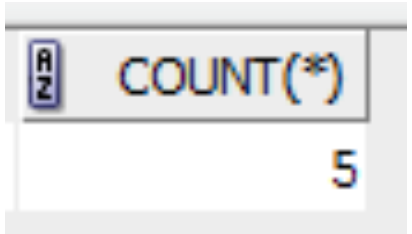
```
select count(remark) from viewing;
```

Answer

- What is the outputs of the following statements:

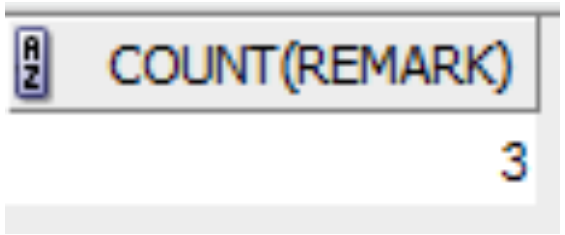
select count(*) from viewing;

Note: COUNT(*) counts all rows of a table, regardless of whether nulls or duplicate values occur.



select count(remark) from viewing;

Note: the NULL values are eliminated in aggregate functions



Example 13 Use of COUNT(*)

- How many properties cost more than £350 per month to rent?

```
SELECT COUNT(*) AS myCount
FROM PropertyForRent
WHERE rent > 350;
```

myCount
5

select propertyNo, rent from propertyForRent;

propertyno character(5)	rent integer
PA14	650
PL94	400
PG4	350
PG36	375
PG21	600
PG16	450

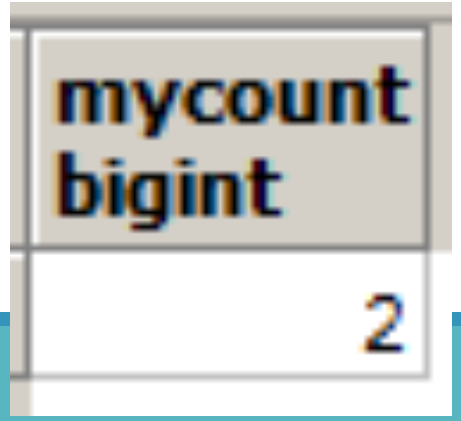
Example 14 Use of COUNT(DISTINCT)

- How many ***different*** properties viewed in May 19?

SELECT COUNT(***DISTINCT*** propertyNo) AS myCount

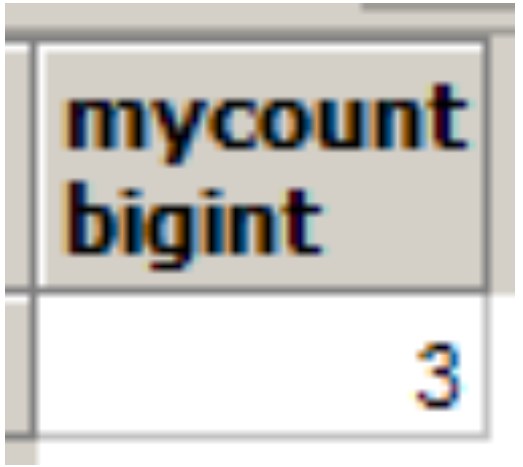
FROM Viewing

WHERE viewDate BETWEEN '1-May-19' AND '31-May-19';



Note: without keyword distinct

SELECT COUNT(propertyNo) AS myCount
FROM Viewing
WHERE viewDate BETWEEN '1-May-19' AND '31-May-19';



Example 15 Use of COUNT and SUM

- Find number of Managers and sum of their salaries.

```
SELECT COUNT(staffNo) AS myCount,
       SUM(salary) AS mySum
FROM Staff
WHERE position = 'Manager';
```

myCount	mySum
2	54000.00

Example 16 Use of MIN, MAX, AVG

- Find minimum, maximum, and average staff salary.

```
SELECT MIN(salary) AS myMin,
       MAX(salary) AS myMax,
       AVG(salary) AS myAvg
FROM Staff;
```

myMin	myMax	myAvg
9000.00	30000.00	17000.00

Reference

- Chapter 6 of Connolly, T and Begg, C, Database Systems: A practical Approach to Design, Implementation, and Management (6th ed.), Boston: Pearson Education.