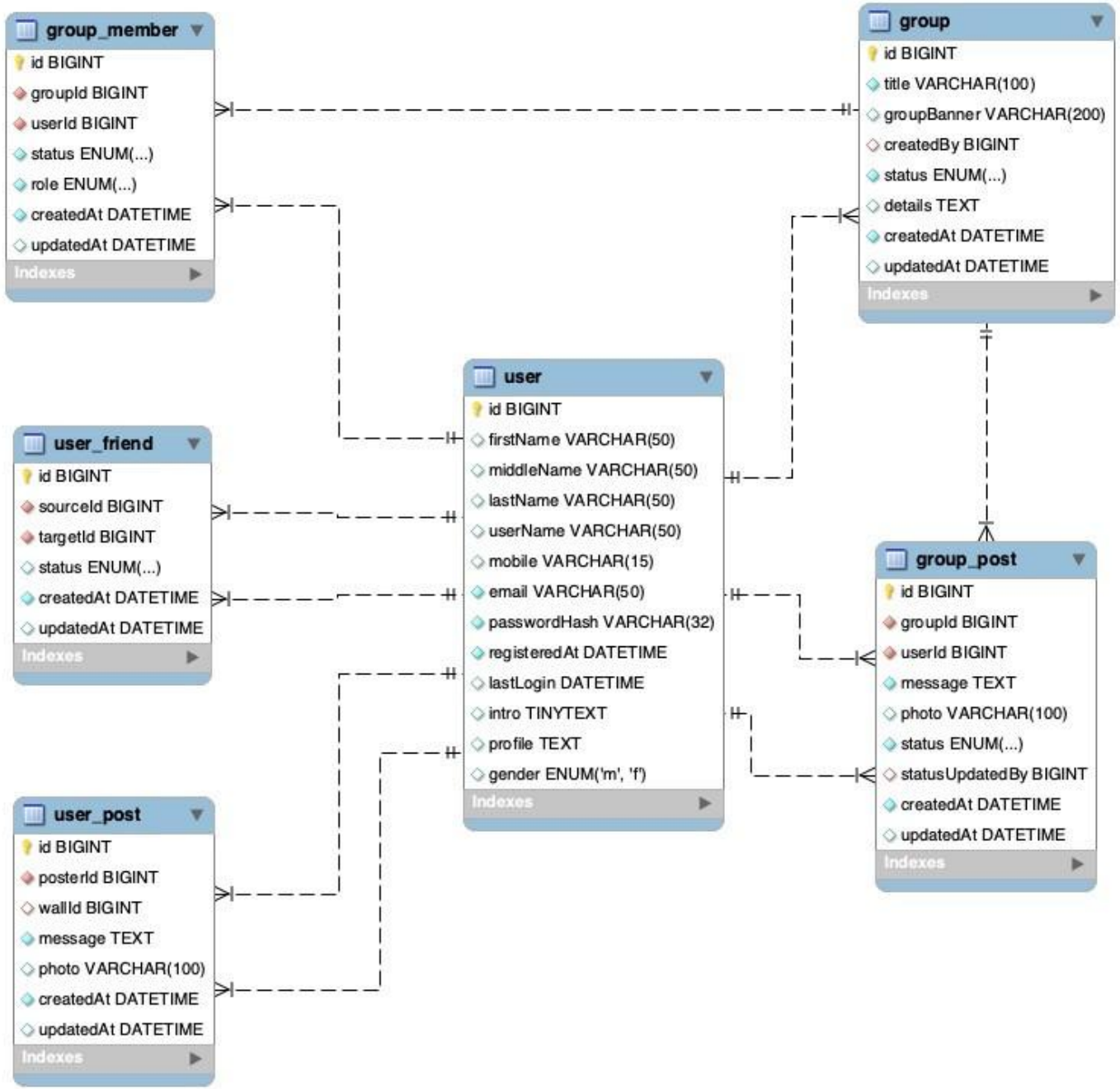
AT82.02

DATA MODELING AND MANAGEMENT

LAB03: SQL (PART II)

DS&AI

Case Study: Social Network System



Lab03: Basic SQL Query

Retrieval
- **SELECT (DISTINCT)**

Filtering Data
- **WHERE** clause

Sorting Data
- **ORDER BY**

Aggregation
- **SUM, COUNT, MIN, MAX, AVG** …
- **HAVING**
- **GROUP BY**

Grouping/Limiting Output
- **LIMIT**

SELECT Syntax

```
SELECT expressions
FROM tables
[WHERE conditions]
[GROUP BY
expressions]  [HAVING
conditions]
[ORDER BY expression [ ASC | DESC
]]  [LIMIT number_rows ];
```

Simple form

```
SELECT expressions

FROM tables;
```

Retrieving Table Info: *

```
SELECT * FROM sns.user;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	firstName	middleName	lastName	userName	mobile	email	passwordHash
1	Kenn	NULL	Naris	knaris0	3186702786	knaris0@skype.com	HcWSiohU72sb
2	Jennifer	NULL	Leehane	jleehane1	6687207855	jleehane1@walmart.com	UzqSWZD5CW6q
3	Merilee	NULL	Cornely	mcornely2	6199838880	mcornely2@cam.ac.uk	96TEt8LdKP
4	Sophronia	Drance	Milton	NULL	5527579377	smilton3@ft.com	wDXJ1EPx
5	Quill	Rendell	Beadell	qbeadell4	6032360300	qbeadell4@salon.com	G2aEss3TrK
6	Silvie	NULL	Vernazza	svernazza5	9787648018	svernazza5@army.mil	kmOGPCum
7	Berton	NULL	Cabell	bcabell6	5197110612	bcabell6@shareasale.com	JsoNM0k
8	Bernete	NULL	Beebe	bbeebe7	4963716518	bbeebe7@bbc.co.uk	YNVVxb
9	Josefa	NULL	Lauxmann	jlauxmann8	8337729135	jlauxmann8@youtu.be	bS69K26nmlKB
10	Lacey	NULL	Bosence	lbosence9	6485032621	lbosence9@seesaa.net	thC0mM0tMTl
11	Ardelia	NULL	Madner	amadnera	1178858385	amadnera@adobe.com	bDPZsyL0

Projection Column list

```
SELECT id,  firstName, lastName, gender FROM
sns.user;
```

id	firstName	lastName	gender
61	Kristofor	Knath	m
62	Nikolas	Ragsdall	NULL
63	Anthony	Croutear	m
64	Harold	Trevain	m
65	Beverly	Agott	f
66	Corrinne	O'Dea	NULL
67	Bobbie	Habbert	m
68	Gregoor	Blueman	m
69	Nichol	Broome	f
70	Nero	Branno	NULL
71	Livy	McCullagh	f

Retrieving Distinct

```
SELECT DISTINCT gender

FROM sns.user;
```

SQL Operators

- Logical Operators
  - **AND OR NOT**
  - **IS (NOT)**
- RANGE Operators
  - **BETWEEN … AND**
  - **(NOT) IN**
- Arithmetic Operators
  - **+, -, *, /**
- Comparison Operators
  - **<, <=, >, >=, =, <> or !=**
- **LIKE** and Wildcard
  - **%** any characters
  - **_** one character

Filtering Data: Single Criteria

```
SELECT id, firstName, lastName, gender

FROM sns.user

WHERE gender = 'f';
```

id	firstName	lastName	gender
2	Jennifer	Leehane	f
9	Josefa	Lauxmann	f
11	Ardelia	Madner	f
19	Evangelia	Macklin	f
25	Codie	Gillespey	f
27	Amil	Stuchbury	f
30	Marena	Metzing	f
36	Moreen	Schoolcroft	f
38	Eirena	Goodwins	f
39	Fredelia	McTrustrie	f
41	Persis	Currao	f
47	Vivyan	Russi	f
48	Ferdina…	Pitkaithly	f
56	Anni	Ather	f
60	Eran	Olle	f
65	Beverly	Agott	f
69	Nichol	Broome	f

Filtering Data: Single Criteria

```
SELECT id, firstName, lastName, gender

FROM sns.user

WHERE gender IS NULL;
```

id	firstName	lastName	gender
1	Kenn	Naris	NULL
3	Merilee	Cornely	NULL
4	Sophronia	Milton	NULL
5	Quill	Beadell	NULL
6	Silvie	Vernazza	NULL
7	Berton	Cabell	NULL
8	Bernete	Beebe	NULL
10	Lacey	Bosence	NULL
12	Tiffany	Schwandner	NULL

Filtering Data: Single Criteria

```
SELECT id, firstName, lastName, gender FROM
sns.user

WHERE gender IS NOT N
```

id	firstName	lastName	gender
2	Jennifer	Leehane	f
9	Josefa	Lauxmann	f
11	Ardelia	Madner	f
16	Samuel	Cutill	m
18	Erwin	Kitchen	m
19	Evangelia	Macklin	f
25	Codie	Gillespey	f
26	Pryce	Southward	m
27	Amil	Stuchbury	f
30	Marena	Metzing	f
31	Ragnar	MacNockater	m

Filtering Data: Range Criteria

```
SELECT id, firstName, lastName, gender
FROM sns.user
WHERE gender = 'm'
      AND id BETWEEN 10 AND 20;
```

	id	firstName	lastName	gender
▸	16	Samuel	Cutill	m
	18	Erwin	Kitchen	m
*	NULL	NULL	NULL	NULL

Filtering Data: IN operators

```
SELECT id, firstName, lastName, gender

FROM sns.user

WHERE firstName IN
('Jennifer','Ardelia','Chase');
```

id	firstName	lastName	gender
2	Jennifer	Leehane	f
11	Ardelia	Madner	f
32	Chase	Anyon	m

In-class exercise (1)

- Select all female users and their contact information.
- Select all users who have not provided their gender and their telephone number.
- Select all users who have registered with in last 7 days. (Using the methods just discussed)

Filtering Data: LIKE and Wildcard

```
SELECT id, firstName

FROM sns.user

WHERE firstName LIKE '%en';
```

id	firstName
36	Moreen
117	Cathyleen
127	Eleen
148	Yasmeen
170	Gayleen
209	Marten
243	Brennen
279	Joleen
NULL	NULL

```
SELECT id, firstName

FROM sns.user

WHERE firstName LIKE 'Jen%';
```

id	firstName
2	Jennifer
78	Jenica
187	Jenna

Filtering Data: LIKE and Wildcard

```
SELECT id, firstName

FROM sns.user

WHERE firstName LIKE '_er%';
```

id	firstName
3	Merilee
7	Berton
8	Bernete
41	Persis
45	Vergil
48	Ferdinande
70	Nero
86	Bernetta
99	Werner
164	Fernande
184	Jeralee

```
SELECT id, firstName

FROM sns.user

WHERE firstName LIKE '%os_';
```

id	firstName
163	Moss

Sorting Data

```
SELECT expressions
FROM tables
[WHERE conditions]
[GROUP BY
expressions]  [HAVING
conditions]
[ORDER BY expression [ ASC | DESC ]]
[LIMIT number_rows ]
```

Sorting Data

ORDER BY sort the output rows

- Sorting the output rows make it easier to read the results
- ORDER BY
  - DESC: high to low ; decreasing order
  - ASC: low to high ; increasing order <- default
- ORDER BY can be used with more than 1 column name
  - Only when the value in the first column is equal, the second column will be used.

Sorting Data

```
SELECT firstName, middleName, lastName FROM sns.user ORDER BY
```

	firstName	middleName	lastName
▸	Abel	Knoles	Janiak
	Adam	Daveridge	Corkan
	Addie	Portugal	Eixenberger
	Adelind	NULL	Towse
	Agathe	NULL	Morehall
	Agnola	NULL	Batisse

```
SELECT firstName, middleName, lastName FROM sns.user ORDER BY firstName
```

	firstName	middleName	lastName
▸	Zondra	Canacott	McNysche
	Zane	NULL	Sends
	Zach	NULL	Keford
	Yasmeen	NULL	Tomblett
	Winston	Parmley	Autie
	Win	NULL	Buncher
	Wes	NULL	Renyard

In-class exercise (2)

- Select users whose name begin with ‘Jo’. Show the output in the descending order using a ‘**`lastName`**’ column.
- Select female users of any first names OR those who haven’t provided their gender with contain ‘*bel*’ in their first name.

Aggregation

SQL provides several AGGREGATE functions that summarize the data rows into a single value. The common aggregate functions are:

- COUNT: Counting satisfied rows
- SUM: Sum value
- AVG: Average value
- MIN: Minimum value
- MAX: Maximum value

Aggregation

```
SELECT COUNT(*) AS   NoOfPost FROM
user_post;
```

	NoOfPost
▶	1000

```
SELECT posterId, COUNT(*) AS
NoOfPost
FROM user_post

GROUP BY posterId;
```

	posterId	NoOfPost
▶	1	3
	2	2
	3	2
	4	6
	5	5
	6	2

Aggregation

```
--Show the average subscription duration of all users

SELECT AVG(TIMESTAMPDIFF(DAY, registeredAt, now()) )
      AS AVGDuration

FROM sns.user;
```

AVGDuration
29.7400

```
-- TIMESTAMPDIFF() returns a different of 2 values of time in the
specific measure (DAY, MONTH, YEAR, ...), ONLY in 'MySQL'
```

In-class exercise (3)

1. Which gender has more average membership duration? Male or female? Show your SQL command that helps you answer.

Limiting output

To limit your output rows, you can use **`LIMIT`** keyword to specify the number of rows to be desired output.

```
-- Select only first 3 rows of users

SELECT * FROM sns.user LIMIT 3;
```

id	firstName	middleName	lastName	userName	mobile	email	passwordHash	registeredAt	lastLogin	intro
1	Kenn	NULL	Naris	knaris0	3186702786	knaris0@skype.com	HcWSiohU72sb	2021-07-09 00:16:09	NULL	Inform
2	Jennifer	NULL	Leehane	jleehane1	6687207855	jleehane1@walmart.com	UzqSWZD5CW6q	2021-08-10 14:29:03	NULL	Progr
3	Merilee	NULL	Cornely	mcornely2	6199838880	mcornely2@cam.ac.uk	96TEt8LdKP	2021-07-11 09:43:31	2021-07-05 03:11:10	Editor
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Limiting output

```
-- Select only first 3 users, starting the second row

SELECT * FROM sns.user LIMIT 1,3;
```

id	firstName	middleName	lastName	userName	mobile	email	passwordHash	registeredAt	lastLogin	intro
2	Jennifer	NULL	Leehane	jleehane1	6687207855	jleehane1@walmart.com	UzqSWZD5CW6q	2021-08-10 14:29:03	NULL	Programr
3	Merilee	NULL	Cornely	mcornely2	6199838880	mcornely2@cam.ac.uk	96TEt8LdKP	2021-07-11 09:43:31	2021-07-05 03:11:10	Editor
4	Sophronia	Drance	Milton	NULL	5527579377	smilton3@ft.com	wDXJ1EPx	2021-08-06 21:07:22	2021-07-12 13:06:18	Desktop S
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Limiting output

```
-- Select top 3 of users who posted the most.
SELECT posterId, COUNT(*) AS NoOfPost

FROM sns.user_post

GROUP BY posterId

ORDER BY NoOfPost DESC

LIMIT 3;
```

posterId	NoOfPost
87	10
257	8
281	8

In-class exercise (4)

1. What are the last 4 groups on the basis of group members? List those 4 groups.
2. Show 3 male and 3 female users who have the longest membership duration. Show your separated SQL commands.

Almost at the end

Some Additional Info…

Attribute Reference

For non-duplicated column name, there is no need to reference table name before attribute name

For duplicated column names, the table name must be referenced before attribute name, e.g.,

```
/* list student names who are in the course 'DMM101'*/
   SELECT firstname, lastname, name /* no need for s.firstname */
   FROM students s, courses c
   WHERE s.course_id =
   c.course_id
   AND s.course_id = 'DMM101' /* s.course_id or c.course_id is required*/
```

SQL Syntax

- **Keyword:** SQL has many keywords that have special meanings such as SELECT, INSERT, UPDATE, DELETE, and DROP. These keywords are the reserved words.
- **Identifiers:** Identifiers refer to specific objects in the database such as tables, columns, indexes, etc. SQL is case-insensitive with respect to keywords and identifiers.
  - Select * From employees;
  - SELECT * FROM EMPLOYEES;
- **Comments:** When parsing SQL, state… ment…

```
SELECT
    employee_id, salary
FROM
    employees
WHERE
    salary < 3000; -- employees with low salary
```

```
/* increase 5% for employees whose salary is less than 3,000 */
UPDATE employees
SET
    salary = salary * 1.05
WHERE
    salary < 3000;
```

References

- SQL Tutorial : https://www.sqltutorial.org/
- SQL in W3School: https://www.w3schools.com/sql/


Thank you.