

Advanced SELECT Queries & Views

Grigorios Loukides
Email: grigorios.loukides@kcl.ac.uk

Session Objectives

2/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

In this session, you will learn:

- More advanced queries
- Views

Table Alias

3/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

- Alias can be used to qualify column names when there is ambiguity
- To use an alias for a table in FROM clause:

`FROM tableName [AS] newName`

Update: Exercise II

4/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested
Readings

Exercises

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

Assume there is a table StaffPropCount that contains names of staff and number of properties they manage:

StaffPropCount(staffNo, fName, lName, propCnt)

Populate StaffPropCount using Staff and PropertyForRent tables

Feature used in Exercise: Table alias.

5/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

FROM Staff same as FROM Staff [AS] S.

UPDATE StaffPropCount same as UPDATE Staff [AS] P.

Useful when you have tables with same columns or long names.

Solution

```
INSERT INTO StaffPropCount (staffNo, fName, IName)
```

```
SELECT S.staffNo, S.fName,S.IName FROM Staff AS S;
```

```
UPDATE StaffPropCount AS S
```

```
SET propCNT=(SELECT COUNT(*)
```

```
FROM PropertyForRent AS P
```

```
WHERE S.StaffNo=P.StaffNo)
```

Alternatively:

```
UPDATE StaffPropCount
```

```
SET propCnt=(SELECT COUNT(*) FROM PropertyForRent
```

```
WHERE StaffPropCount.StaffNo=PropertyForRent.StaffNo)
```

Data Manipulation Language

6/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

DML allows to retrieve and update data:

- SELECT statement retrieves data
- INSERT, UPDATE, DELETE statements update data

SELECT ...

LIMIT {rowCount | rowCount OFFSET offset}

- The LIMIT clause can be used to constrain the number of rows returned by the SELECT statement.
- The OFFSET clause can be used together with the LIMIT to specify the offset of the first row to return

LIMIT

8/64

Introduction

Data
Retrieval

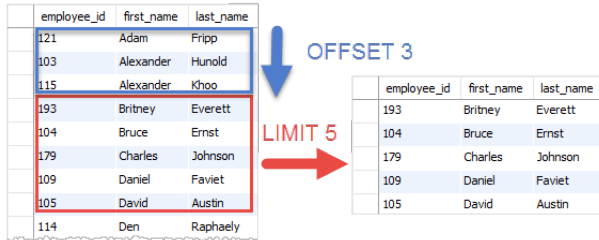
LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises



```
SELECT *  
FROM EMPLOYEE  
ORDER BY employee_id  
LIMIT 5 OFFSET 3;
```


Subqueries

9/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

- Some SQL statements can have a SELECT embedded within them
- A subselect can be used in WHERE and HAVING clauses of an outer SELECT, where it is called a subquery or nested query:
 - Subquery produce a temporary table with results that can be accessed by the outer statement
 - Subqueries can be used following a relational operator (=, <, >, <=, >=, <>) in WHERE and HAVING clauses
 - Subqueries are always enclosed by parentheses
- Subselects may also appear in INSERT, UPDATE, and DELETE statements

Subqueries: Example

10/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

List staff who work in branch at '163 Main St'.

```
SELECT staffNo, fName, lName, position
FROM Staff
WHERE branchNo =
      (SELECT branchNo
       FROM Branch
       WHERE street = '163 Main St');
```

Subqueries: Example

11/64

```
SELECT staffNo, fName, lName, position
FROM Staff
WHERE branchNo =
    (SELECT branchNo
     FROM Branch
     WHERE street = '163 Main St');
```

- Inner SELECT finds branch number for branch at '163 Main St' ('B003').
- Outer SELECT then retrieves details of all staff who work at this branch.

staffNo	fName	lName	position
SG37	Ann	Beech	Assistant
SG14	David	Ford	Supervisor
SG5	Susan	Brand	Manager

Subquery Rules

12/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

- ORDER BY clause may not be used in a subquery (although it may be used in outermost SELECT)
- Subquery SELECT list must consist of a single column name or expression, except for subqueries that use EXISTS
- By default, column names refer to table name in FROM clause of subquery. Can refer to a table in FROM using an alias
- When subquery is an operand in a comparison, subquery must appear on right-hand side
- A subquery may not be used as an operand in an expression

Subqueries & IN

13/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

- IN to select rows when the value in the expression must be among the values returned by the subquery
- NOT IN to select rows when the value is not among the values returned by the subquery

Subqueries: Example

14/64

Introduction

Data
Retrieval

LIMIT
Subqueries

Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

```
SELECT propertyNo, street, city, postcode, type,  
       rooms, rent  
FROM PropertyForRent  
WHERE staffNo IN  
       (SELECT staffNo  
        FROM Staff  
        WHERE branchNo = 'B02')
```

- What does this query do?

- ANY and ALL may be used with subqueries that produce a single column of numbers
- ALL is true if satisfied by **all** values produced by subquery
- ANY is true if satisfied by **any** values produced by subquery
- If subquery is empty, ALL returns true, ANY returns false

ALL/ANY: Example

16/64

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

```
SELECT staffno, fname, lname, position, salary
FROM staff
WHERE salary > ALL (SELECT salary
                     FROM staff
                     WHERE branchno = 'B003')
```

- What does this query do?

EXISTS

17/64

Introduction

Data
Retrieval

LIMIT
Subqueries

Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

- EXISTS and NOT EXISTS are for use only with subqueries
- Produce a simple true/false result
- True if and only if there exists **at least one** row in result table returned by subquery
- False if subquery returns an empty result table

- NOT EXISTS is the opposite of EXISTS
- As EXISTS (or NOT EXISTS) checks only for existence (or non-existence) of rows in the subquery's result table, the subquery can contain any number of columns

EXISTS / NOT EXISTS: Example

18/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

```
SELECT staffNo, fName, lName, position
FROM Staff AS s
WHERE EXISTS
    (SELECT *
     FROM Branch AS b
     WHERE s.branchNo = b.branchNo AND
           city = 'London')
```

- What does this query do?

Subqueries in FROM

19/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

Subqueries are legal in a SELECT statement's FROM clause.
The actual syntax is:

```
SELECT ... FROM (subquery) [AS] name ...
```

- The **name** is mandatory, because every table in a FROM clause must have a name
- Any columns in the subquery select list must have unique names

Subqueries in FROM: Example

20/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

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

```
SELECT MAX(staffCount),branchNo
FROM (SELECT COUNT(staffNo) AS staffCount,branchNo
      FROM staff GROUP BY branchNo) AS staffInBranch;
```

What does this query do?

What does this query do?

```
SELECT sub.*  
  FROM (  
    SELECT *  
      FROM tutorial.sf_crime_incidents_2014_01  
     WHERE day_of_week = 'Friday'  
  ) sub  
WHERE sub.resolution = 'NONE'
```

Multi-Table Queries

22/64

Introduction

Data
Retrieval

LIMIT
Subqueries

Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

- If result columns come from more than one table must use a join
- To perform join, include more than one table in FROM clause
- Use comma as separator and typically include WHERE clause to specify join column(s)

Multi-Table Queries: Simple Example

23/64

Table : Patient

idPatient	name	age	sex	idDoctor
1	Tom	36	M	1
2	Sam	55	M	NULL
3	Marie	74	F	2

Table : Doctor

idDoctor	name
1	John Smith
2	Anne Smith
3	Julie Smith

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

Multi-Table Queries: Simple Example

24/64

```
SELECT *  
FROM Patient, Doctor;
```

idPatient	name	age	sex	idDoctor	idDoctor	name
1	Tom	36	M	1	1	John Smith
2	Sam	55	M	NULL	1	John Smith
3	Marie	74	F	2	1	John Smith
1	Tom	36	M	1	2	Anne Smith
2	Sam	55	M	NULL	2	Anne Smith
3	Marie	74	F	2	2	Anne Smith
1	Tom	36	M	1	3	Julie Smith
2	Sam	55	M	NULL	3	Julie Smith
3	Marie	74	F	2	3	Julie Smith

Introduction

Data
Retrieval

LIMIT
Subqueries

Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

Multi-Table Queries: Example

25/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

```
SELECT *  
FROM Patient P, Doctor D  
WHERE D.idDoctor = P.idDoctor;
```

- What does this query do?

Multi-Table Queries: Example

26/64

Introduction

Data
Retrieval

LIMIT
Subqueries

Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

```
SELECT *  
FROM Patient P, Doctor D  
WHERE D.idDoctor = P.idDoctor;
```

idPatient	name	age	sex	idDoctor	idDoctor	name
1	Tom	36	M	1	1	John Smith
3	Marie	74	F	2	2	Anne Smith

Join

27/64

Introduction

Data
Retrieval

LIMIT
Subqueries

Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

Join

A Join operation is used to combine rows from two or more tables, based on a common field between them

<i>T</i>		<i>U</i>				
<i>A</i>	<i>B</i>	<i>B</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>C</i>
<i>a</i>	1	1	<i>x</i>	<i>a</i>	1	<i>x</i>
<i>b</i>	2	1	<i>y</i>	<i>a</i>	1	<i>y</i>
		3	<i>z</i>			



- SQL provides alternative ways to specify joins between tables:
 - 1 FROM Patient P, Doctor D
WHERE D.idDoctor = P.idDoctor;
 - 2 FROM Patient P JOIN Doctor D
ON D.idDoctor = P.idDoctor;
 - 3 FROM Patient JOIN Doctor USING(idDoctor);
 - 4 FROM Patient NATURAL JOIN Doctor
- In each case, FROM replaces original FROM and WHERE. However, the 2nd way produces table with two identical clientNo columns

JOIN ON

29/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

```
FROM Patient P JOIN Doctor D  
      ON D.idDoctor = P.idDoctor;
```

The ON clause determines the condition for making the join

idPatient	name	age	sex	idDoctor	idDoctor	name
1	Tom	36	M	1	1	John Smith
3	Marie	74	F	2	2	Anne Smith

JOIN USING

30/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

```
FROM Patient JOIN Doctor USING(idDoctor);
```

The `USING(column_list)` clause names a list of columns that must exist in both tables

idDoctor	idPatient	name	age	sex	name
1	1	Tom	36	M	John Smith
2	3	Marie	74	F	Anne Smith

NATURAL JOIN

31/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

FROM Patient NATURAL JOIN Doctor

The NATURAL JOIN of two tables is defined to be semantically equivalent to an JOIN with a USING clause that names all columns that exist in both tables

idDoctor	idPatient	name	age	sex
----------	-----------	------	-----	-----

Why the result is empty??

Outer Joins

32/64

Introduction

Data
Retrieval

LIMIT
Subqueries

Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

Often in joining two tables, a row in one table does not have a matching row in the other table; in other words, there is no matching value in the join columns.

Outer Join

Return all rows from at least one of the tables even when there are no matching values in the other table

Types of Outer Joins

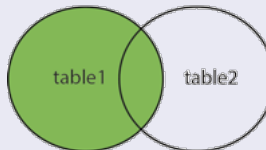
33/64

Left Join

Returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match.

```
SELECT column_name(s)
FROM table1 LEFT [OUTER] JOIN table2
    ON table1.column_name=table2.column_name;
```

LEFT JOIN



Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

Left Join: Example I

34/64

Introduction

Data
Retrieval

LIMIT
Subqueries

Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

a_id	name
1	apple
2	orange
3	tomato
4	cucumber

b_id	name
A	apple
B	banana
C	cucumber
D	dill

```
SELECT *  
FROM TableA  
LEFT OUTER JOIN TableB  
ON tableA.name = tableB.name;
```

a_id	name
1	apple
2	orange
3	tomato
4	cucumber

b_id	name
A	apple
null	null
null	null
B	banana
C	cucumber
D	dill

a_id	TableA.name	b_id	TableB.name
1	apple	A	apple
2	orange	null	null
3	tomato	null	null
4	cucumber	C	cucumber

Left Join: Example II

35/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

Table : Patient

idPatient	name	age	sex	idDoctor
1	Tom	36	M	1
2	Sam	55	M	NULL
3	Marie	74	F	2

Table : Doctor

idDoctor	name
1	John Smith
2	Anne Smith
3	Julie Smith

```
SELECT *  
FROM Patient P LEFT JOIN Doctor D ON D.idDoctor=P.idDoctor;
```

Left Join: Example II

36/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested
Readings

Exercises

```
SELECT *  
FROM Patient P LEFT JOIN Doctor D  
ON D.idDoctor=P.idDoctor;
```

idPatient	name	age	sex	idDoctor	idDoctor	name
1	Tom	36	M	1	1	John Smith
3	Marie	74	F	2	2	Anne Smith
2	Sam	55	M	NULL	NULL	NULL

Types of Outer Joins

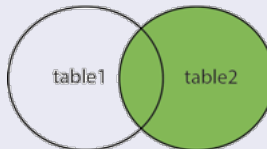
37/64

Right Join

Returns all rows from the right table (table2), with the matching rows in the left table (table1). The result is NULL in the left side when there is no match

```
SELECT column_name(s)
FROM table1 RIGHT [OUTER] JOIN table2
    ON table1.column_name=table2.column_name;
```

RIGHT JOIN



Right Join: Example I

38/64

Introduction

Data
Retrieval

LIMIT
Subqueries

Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

a_id	name
1	apple
2	orange
3	tomato
4	cucumber

b_id	name
A	apple
B	banana
C	cucumber
D	dill

```
SELECT *  
FROM TableA  
RIGHT OUTER JOIN TableB  
ON tableA.name = tableB.name;
```

a_id	name
1	apple
<i>null</i>	<i>null</i>
2	orange
3	tomato
4	cucumber
<i>null</i>	<i>null</i>

b_id	name
A	apple
B	banana
C	cucumber
D	dill

a_id	TableA.name	b_id	TableB.name
1	apple	A	apple
<i>null</i>	<i>null</i>	B	banana
4	cucumber	C	cucumber
<i>null</i>	<i>null</i>	D	dill

Right Join: Example II

39/64

Table : Patient

idPatient	name	age	sex	idDoctor
1	Tom	36	M	1
2	Sam	55	M	NULL
3	Marie	74	F	2

Table : Doctor

idDoctor	name
1	John Smith
2	Anne Smith
3	Julie Smith

```
SELECT *  
FROM Patient P RIGHT JOIN Doctor D ON D.idDoctor=P.idDoctor;
```

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

Right Join: Example II

40/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

```
SELECT *  
FROM Patient P RIGHT JOIN Doctor D  
ON D.idDoctor=P.idDoctor;
```

idPatient	name	age	sex	idDoctor	idDoctor	name
1	Tom	36	M	1	1	John Smith
3	Marie	74	F	2	2	Anne Smith
NULL	NULL	NULL	NULL	NULL	3	Julie Smith

Types of Outer Joins

41/64

Introduction

Data
Retrieval

LIMIT
Subqueries

Joins
Union
Operation

View
Management

Conclusion

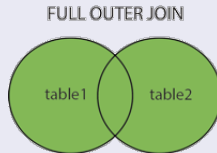
Suggested
Readings

Exercises

Full Join

The FULL OUTER JOIN keyword returns all rows from the left table (table1) and from the right table (table2).

```
SELECT column_name(s)
FROM table1 FULL [OUTER] JOIN table2
    ON table1.column_name=table2.column_name;
```



MySQL does not have this command!!

Full Join: Example I

42/64

Introduction

Data
Retrieval

LIMIT
Subqueries

Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

TableA

a_id	name
1	apple
2	orange
3	tomato
4	cucumber

TableB

b_id	name
A	apple
B	banana
C	cucumber
D	dill

```
SELECT *  
FROM TableA  
FULL OUTER JOIN TableB  
ON tableA.name = tableB.name;
```

TableA

a_id	name
1	apple
<i>null</i>	<i>null</i>
2	orange
3	tomato
4	cucumber
<i>null</i>	<i>null</i>

TableB

b_id	name
A	apple
B	banana
<i>null</i>	<i>null</i>
<i>null</i>	<i>null</i>
C	cucumber
D	dill

a_id	TableA.name	b_id	TableB.name
1	apple	A	apple
<i>null</i>	<i>null</i>	B	banana
2	orange	<i>null</i>	<i>null</i>
3	tomato	<i>null</i>	<i>null</i>
4	cucumber	C	cucumber
<i>null</i>	<i>null</i>	D	dill

Union Operation

43/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins

Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

```
SELECT ...  
UNION [ALL | DISTINCT]  
SELECT ...
```

- UNION is used to combine the result from multiple SELECT statements into a single result set
- The column names from the first SELECT statement are used as the column names for the results returned
- Selected columns listed in corresponding positions of each SELECT statement should have the same data type.
- If ALL specified, result can include duplicate rows

UNION: Example

44/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins

Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

```
(SELECT city
FROM Branch
WHERE city IS NOT NULL)
UNION
(SELECT city
FROM PropertyForRent
WHERE city IS NOT NULL)
```

- What does this query do?

Full Join: Using Union

45/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins

Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

idPatient	name	age	sex	idDoctor	idDoctor	name
1	Tom	36	M	1	1	John Smith
2	Sam	55	M	NULL	2	Anne Smith
3	Marie	74	F	2	3	Julie Smith

```
SELECT *  
FROM Patient P FULL JOIN Doctor D ON D.idDoctor=P.idDoctor;
```

idPatient	name	age	sex	idDoctor	idDoctor	name
1	Tom	36	M	1	1	John Smith
3	Marie	74	F	2	2	Anne Smith
2	Sam	55	M	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	3	Julie Smith

How can we obtain this result without using FULL JOIN??

- A view is a virtual table representing a subset of columns and/or rows and/or column expressions from one or more base tables or views
- A view is created using the CREATE VIEW statement which is a DDL statement
- Views can be used to simplify the structure of the database and make queries easier to write. They can also be used to protect certain columns and/or rows from unauthorized access.

- Dynamic result of one or more relational operations operating on base relations to produce another relation.
- Virtual table that does not necessarily actually exist in the database but is produced upon request, at time of request.
- Allows each user to have his or her own view of the database

View Benefits

48/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

**View
Management**

Conclusion

Suggested
Readings

Exercises

- Reduce complexity
- Provide a level of security
- Provide a mechanism to customize the appearance of the database
- Present a consistent, unchanging picture of the structure of the database, even if the underlying database is changed

Create View

49/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

```
CREATE VIEW viewName [(columnList)] AS selectStatement  
[WITH CHECK OPTION]
```

Creates a new view

- The *selectStatement* is a SELECT statement that provides the definition of the view.
- The view definition is “frozen” at creation time and is not affected by subsequent changes to the definitions of the underlying tables
- To define names for the view columns, the optional *columnList* clause can be given as a list

Create View: Exercise

50/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

**View
Management**

Conclusion

Suggested
Readings

Exercises

Create view so that manager at branch B003 can only see details for staff who work in his or her office.

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

Updatable Views

51/64

```
CREATE VIEW viewName [(columnList)] AS selectStatement  
[WITH CHECK OPTION]
```

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

Some views are updatable

- Some views are updatable. That is, you can use them to update the contents of the underlying table
- For a view to be updatable, there must be a one-to-one relationship between the rows in the view and the rows in the underlying table.
 - The WITH CHECK OPTION clause prevent inserts to rows for which the WHERE clause in the `selectStatement` is not true. It also prevents updates to rows for which the WHERE clause is true but the update would cause it to be not true (in other words, it prevents visible rows from being updated to nonvisible rows).

Updatable Views

52/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

```
CREATE VIEW Manager3Staff
AS SELECT *
FROM Staff
WHERE branchNo = B003
WITH CHECK OPTION;
```

Given the view above, what happens with the following update?

```
UPDATE Manager3Staff
SET branchNo = B005
WHERE staffNo = SG37;
```

Drop View

53/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

```
DROP VIEW [IF EXISTS] viewName [, viewName]
```

Removes one or more views

Conclusion

54/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

In this session we have covered:

- SELECT
 - Subqueries, Joins, Union
- View Management

Suggested Readings

55/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

**Suggested
Readings**

Exercises

- Chapters 4 and 5 of Fundamentals of Database Systems. Elmasri & Navathe.
- Chapters 5 and 6 of Database systems: a practical approach to design, implementation, and management. Connolly, Thomas M; Begg, Carolyn

Exercise I

56/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

- List the staff name and salary of the member of the staff managing the most expensive flat

Subqueries: Exercise I

57/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

- List properties handled by staff at '163 Main St'.

Subqueries: Exercise II

58/64

Introduction

Data
Retrieval

LIMIT
Subqueries
Joins
Union
Operation

View
Management

Conclusion

Suggested
Readings

Exercises

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

- List all staff whose salary is greater than the average salary

Subqueries: Exercise III

59/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

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

- List all staff whose salary is greater than the average salary, and show by how much

Subqueries: Exercise IV

60/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested
Readings

Exercises

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

- List the name of members of the staff who do not manage any property

Join: Exercise I

61/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

- For each branch, list numbers and names of staff who manage properties, and properties they manage

Join: Exercise II

62/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

- For each branch, list staff who manage properties, including city in which branch is located and properties they manage

Join: Exercise III

63/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

- List the staff names and surnames together with the number of properties handled by each staff member

Outer Join: Exercise I

64/64

Introduction

Data

Retrieval

LIMIT

Subqueries

Joins

Union

Operation

View

Management

Conclusion

Suggested

Readings

Exercises

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

- List the staff names and surnames together with the number of properties handled by each staff member