IS475/675

Agenda:
03/17/2025

Answer

- Review Test 1
- Answer questions about HW#5

Show

- Show additional DDL statements

Learn

- Learn how to write the SELECT statement
  - Structure and order of execution
  - Output
  - Functions

Structure of first test

Component	Questions	Percentage
Multiple Choice	27 questions	40%
Skeleton ERD – fill in relationships, foreign keys, additional attributes	1 question	20%
Database Design – create an ERD from specifications	1 question	40%

Sample calculation with sample scores below. Test is worth 200 points (20% of final grade for class)

Multiple choice: score of 18. 18*1.5 = 27% out of 100
Skeleton ERD = score of 18. 18%
Database Design = score of 32. 32%
Test score = 77%

Total points = .77 * 200 = 154

Results from the test

- Average: 81.2%

Test Score Range	Count of Tests
>= 90%	27
< 90% and >= 80%	20
> 80% and <= 70%	13
> 70% and <= 60%	13
< 60%	6

Class structure to learn SQL

- **SQL Lab Exercises** will help you learn how to program in SQL. These should be done in class on lab days, but could be done on your own time if you prefer.
- We will do part of the exercises together, but you can complete them from the exercise document provided on WebCampus.
- Exercises are not graded but are necessary to do the assignments.
- You will complete **homework assignments** to help challenge your skills – they will go beyond what is done in the exercises.
  - HW#5: Build a set of 10 interrelated tables that will be used for the remaining **homework assignments**.
  - HW#6: Write SQL SELECT statements with a single underlying table.
  - HW#7: Write SQL SELECT statements with multiple underlying tables.
  - HW#8: Write complex SQL SELECT statements.

SQL keyword classification

Data Definition Commands (DDL)	Data Manipulation Commands (DML)	Data Control Commands (DCL)
CREATE	INSERT	GRANT
ALTER	UPDATE	REVOKE
DROP	DELETE	COMMIT
	TRUNCATE	ROLLBACK
	SELECT	SET TRANSACTION

ALTER changes the structure of a database object: examples of ALTER

```
ALTER TABLE     tblVendor
ADD             FirstBuyDate DATETIME;
```

```
ALTER TABLE     tblVendorReview
ALTER COLUMN    ReviewComments          VARCHAR(500);
```

```
ALTER TABLE     tblVendorReview
ADD FOREIGN KEY (ProductID)
REFERENCES      tblProduct(ProductID);
```

DELETE removes one or more rows from a table: Examples of DELETE

```
DELETE FROM     tblPurchaseOrder
WHERE           OrderID = '045687';
```

```
DELETE FROM     tblPurchaseOrder
WHERE   vendorID = '20566'
AND     month(PODatePlaced) = 3;
```

```
DELETE FROM     tblPurchaseHistory
WHERE           productID = 'G0983';
```

Update changes the contents of a column in one or more rows in a table: Examples of UPDATE

```
UPDATE  tblPurchaseHistory
SET     VendorID = '87333'
WHERE   ProductID = '87622';
```

```
UPDATE  tblReceiver
SET     qtyreceived = 23
WHERE   DateReceived = '02/15/2025'
AND     qtyreceived = 35
```

```
UPDATE  tblVendor
SET     VendorState = UPPER(VendorState);
```

Getting data from a database

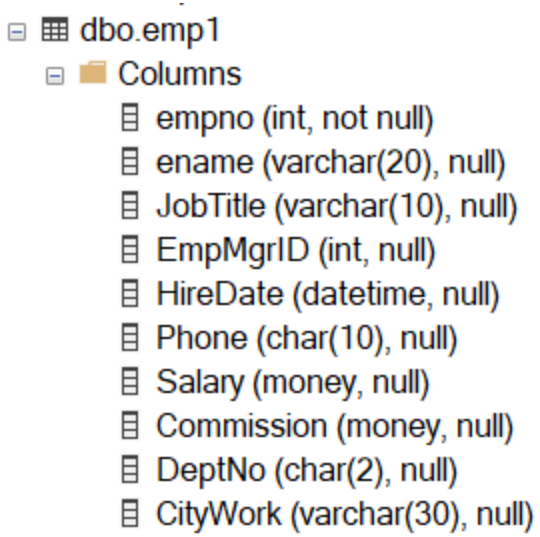
```
SELECT      [all or distinct] (what columns)
FROM        (table or tables)
WHERE       (condition for each row)
GROUP BY    (grouping fields)
HAVING      (condition for the group)
ORDER BY    (sort fields)
```

Most often referred to as a SQL “Query”

Processing by SQL Select Statement

- SQL Select statements produce a result table which can be used by an application and/or user interface program.
- The result table is temporary.
- The result table is created from data stored in permanent database objects (usually tables).
- A SQL Select reads the tables that are used to create the result table one row at a time – there is an "implied loop" that reads through every single row in the underlying table to create the result table.
- The result table is built in a series of steps based on the actual order of execution of the Select statement.

```
SELECT      *
FROM        emp1;
```



	EmpNo	Ename	JobTitle	EmpMgrID	HireDate	Phone	Salary	Commission	DeptNo	CityWork
1	7839	KING, MARGARET	President	NULL	2023-11-17 00:00:00.000	7757845611	9000.00	NULL	10	Reno
2	8698	NG, JUNE	Clerk	7839	2019-06-15 00:00:00.000	7754562501	3120.00	NULL	10	San Diego
3	7782	CLARK, ROBERT	MANAGER	7839	1999-06-09 00:00:00.000	7759810021	2450.00	7230.00	10	San Francisco
4	7566	JONES, MARTIN	MANAGER	7839	2018-04-02 00:00:00.000	8056719332	2975.00	NULL	20	Reno
5	7654	MARTIN, WILLIAM	SALESMAN	7698	2023-09-28 00:00:00.000	8586712300	5250.00	3400.00	30	Reno
6	7499	ALLEN, BERTRAM	SALESMAN	7698	2025-03-20 00:00:00.000	8586723441	2600.00	3300.00	30	Las Vegas
7	7844	TURNER, ELIZABETH	SALESMAN	7698	2024-09-08 00:00:00.000	7754519002	6000.00	7500.00	33	Reno
8	7900	JAMES, KATHERINE	CLERK	7698	2024-10-01 00:00:00.000	7875623456	1950.00	8000.00	30	San Bernardino
9	7902	WONG, BRADFORD	ANALYST	7566	2025-03-07 00:00:00.000	9098337788	5500.00	3500.00	20	San Francisco
10	7521	WARD, ROBERT	SALESMAN	7698	2022-02-22 00:00:00.000	8056671223	3250.00	5000.00	30	Las Vegas
11	9015	JOHNSON, JAMES	SALESMAN	7698	2017-12-15 00:00:00.000	8058912334	2900.00	5000.00	30	Reno
12	8015	MARTINEZ, CONSUELO	ANALYST	7566	2024-12-15 00:00:00.000	8058938924	3900.00	5000.00	20	Reno
13	6743	QUESTA, MARIA	SALESMAN	7566	2025-02-12 00:00:00.000	7758891111	3824.00	6200.00	30	San Diego
14	6011	MARQUEZ, JOTEN	CLERK	7698	2025-02-05 00:00:00.000	7756611255	2150.00	NULL	10	San Diego
15	7788	CHENG, SUN-LIN	ANALYST	7566	2024-09-23 00:00:00.000	7756772990	3900.00	NULL	20	Reno

Statement Component	Explanation
SELECT	List which columns should be in the result table. Can use functions, do calculations, create new fields, use conditional logic.
FROM	List which database tables are used to create the result table. This is where you will join tables together based on their foreign keys.
WHERE	Put condition(s) to determine what rows from the database tables should be included in the result table.
GROUP BY	Establish fields used to group/aggregate the data.
HAVING	Put condition(s) to determine which groups should be included in the result table.
ORDER BY	Sort the result table.

More about the SQL Select

- Can do calculations.
- Can use functions.
- Can do aggregations of data (one row in the result table = multiple rows in the underlying table).
- Can do conditional logic.
- Can be 1000's of lines long.
- Can contain multiple tables.

Control the columns on the result table

```
SELECT ename,
       salary,
       commission,
       hiredate
FROM   emp1;
```

	ename	salary	commission	hiredate
1	KING, MARGARET	9000.00	NULL	2023-11-17 00:00:00.000
2	NG, JUNE	3120.00	NULL	2019-06-15 00:00:00.000
3	CLARK, ROBERT	2450.00	7230.00	1999-06-09 00:00:00.000
4	JONES, MARTIN	2975.00	NULL	2018-04-02 00:00:00.000
5	MARTIN, WILLIAM	5250.00	3400.00	2023-09-28 00:00:00.000
6	ALLEN, BERTRAM	2600.00	3300.00	2025-03-20 00:00:00.000
7	TURNER, ELIZABETH	6000.00	7500.00	2024-09-08 00:00:00.000
8	JAMES, KATHERINE	1950.00	8000.00	2024-10-01 00:00:00.000
9	WONG, BRADFORD	5500.00	3500.00	2025-03-07 00:00:00.000
10	WARD, ROBERT	3250.00	5000.00	2022-02-22 00:00:00.000
11	JOHNSON, JAMES	2900.00	5000.00	2017-12-15 00:00:00.000
12	MARTINEZ, CONSUELO	3900.00	5000.00	2024-12-15 00:00:00.000
13	QUESTA, MARIA	3824.00	6200.00	2025-02-12 00:00:00.000
14	MARQUEZ, JOTEN	2150.00	NULL	2025-02-05 00:00:00.000
15	CHENG, SUN-LIN	3900.00	NULL	2024-09-23 00:00:00.000

What is a Calculation?

```
SELECT  ename,
        salary,
        commission,
        salary + commission,
        (salary + commission) * 1.05
FROM    emp1;
```

	ename	salary	commission	(No column name)	(No column name)
1	KING, MARGARET	9000.00	NULL	NULL	NULL
2	NG, JUNE	3120.00	NULL	NULL	NULL
3	CLARK, ROBERT	2450.00	7230.00	9680.00	10164.000000
4	JONES, MARTIN	2975.00	NULL	NULL	NULL
5	MARTIN, WILLIAM	5250.00	3400.00	8650.00	9082.500000
6	ALLEN, BERTRAM	2600.00	3300.00	5900.00	6195.000000
7	TURNER, ELIZABETH	6000.00	7500.00	13500.00	14175.000000
8	JAMES, KATHERINE	1950.00	8000.00	9950.00	10447.500000
9	WONG, BRADFORD	5500.00	3500.00	9000.00	9450.000000
10	WARD, ROBERT	3250.00	5000.00	8250.00	8662.500000
11	JOHNSON, JAMES	2900.00	5000.00	7900.00	8295.000000
12	MARTINEZ, CONSUELO	3900.00	5000.00	8900.00	9345.000000
13	QUESTA, MARIA	3824.00	6200.00	10024.00	10525.200000
14	MARQUEZ, JOTEN	2150.00	NULL	NULL	NULL
15	CHENG, SUN-LIN	3900.00	NULL	NULL	NULL

What is a column alias?

```
SELECT  ename,
        salary,
        commission,
        salary + commission  TotalPay,
        (salary + commission) * 1.05 TotalPayUp5Percent
FROM    emp1;
```

	ename	salary	commission	TotalPay	TotalPayUp5Percent
1	KING, MARGARET	9000.00	NULL	NULL	NULL
2	NG, JUNE	3120.00	NULL	NULL	NULL
3	CLARK, ROBERT	2450.00	7230.00	9680.00	10164.000000
4	JONES, MARTIN	2975.00	NULL	NULL	NULL
5	MARTIN, WILLIAM	5250.00	3400.00	8650.00	9082.500000
6	ALLEN, BERTRAM	2600.00	3300.00	5900.00	6195.000000
7	TURNER, ELIZABETH	6000.00	7500.00	13500.00	14175.000000
8	JAMES, KATHERINE	1950.00	8000.00	9950.00	10447.500000
9	WONG, BRADFORD	5500.00	3500.00	9000.00	9450.000000
10	WARD, ROBERT	3250.00	5000.00	8250.00	8662.500000
11	JOHNSON, JAMES	2900.00	5000.00	7900.00	8295.000000
12	MARTINEZ, CONSUELO	3900.00	5000.00	8900.00	9345.000000
13	QUESTA, MARIA	3824.00	6200.00	10024.00	10525.200000
14	MARQUEZ, JOTEN	2150.00	NULL	NULL	NULL
15	CHENG, SUN-LIN	3900.00	NULL	NULL	NULL

What is a function?

```
SELECT   LOWER(ename) EmployeeName,
         salary,
         commission,
         salary + ISNULL(commission,0)  TotalPay,
         (salary + ISNULL(commission,0)) * 1.05   TotalPayUp5Percent
FROM    emp1;
```

	EmployeeName	salary	commission	TotalPay	TotalPayUp5Percent
1	king, margaret	9000.00	NULL	9000.00	9450.000000
2	ng, june	3120.00	NULL	3120.00	3276.000000
3	clark, robert	2450.00	7230.00	9680.00	10164.000000
4	jones, martin	2975.00	NULL	2975.00	3123.750000
5	martin, william	5250.00	3400.00	8650.00	9082.500000
6	allen, bertram	2600.00	3300.00	5900.00	6195.000000
7	turner, elizabeth	6000.00	7500.00	13500.00	14175.000000
8	james, katherine	1950.00	8000.00	9950.00	10447.500000
9	wong, bradford	5500.00	3500.00	9000.00	9450.000000
10	ward, robert	3250.00	5000.00	8250.00	8662.500000
11	johnson, james	2900.00	5000.00	7900.00	8295.000000
12	martinez, consuelo	3900.00	5000.00	8900.00	9345.000000
13	questa, maria	3824.00	6200.00	10024.00	10525.200000
14	marquez, joten	2150.00	NULL	2150.00	2257.500000
15	cheng, sun-lin	3900.00	NULL	3900.00	4095.000000

Filter the rows on the result table

```
SELECT   ename,
         salary,
         deptno
FROM    emp1
WHERE salary < 3000;
```

	ename	salary	deptno
1	CLARK, ROBERT	2450.00	10
2	JONES, MARTIN	2975.00	20
3	ALLEN, BERTRAM	2600.00	30
4	JAMES, KATHERINE	1950.00	30
5	JOHNSON, JAMES	2900.00	30
6	MARQUEZ, JOTEN	2150.00	10

```
SELECT   ename,
         salary,
         deptno
FROM    emp1
WHERE salary < 3000 and deptno = '30';
```

	ename	salary	deptno
1	ALLEN, BERTRAM	2600.00	30
2	JAMES, KATHERINE	1950.00	30
3	JOHNSON, JAMES	2900.00	30

```
SELECT   ename,
         salary,
         deptno
FROM    emp1
WHERE salary < 3000 or deptno = '30'
ORDER BY salary;
```

	ename	salary	deptno
1	CLARK, ROBERT	2450.00	10
2	JONES, MARTIN	2975.00	20
3	MARTIN, WILLIAM	5250.00	30
4	ALLEN, BERTRAM	2600.00	30
5	JAMES, KATHERINE	1950.00	30
6	WARD, ROBERT	3250.00	30
7	JOHNSON, JAMES	2900.00	30
8	QUESTA, MARIA	3824.00	30
9	MARQUEZ, JOTEN	2150.00	10

Structure of the SELECT statement

```
SELECT      [all or distinct]
FROM        (table)
WHERE       (condition)
GROUP BY    (grouping fields)
HAVING      (condition)
ORDER BY    (sort fields)
```

Referred to as the "SELECT LIST"

Order of Actual Execution:

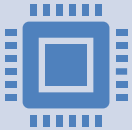
1) FROM
2) WHERE
3) GROUP BY
4) HAVING
5) SELECT
6) ORDER BY

When a SELECT statement is executed, the result is referred to as a "result table". It is a memory-based table.

How to learn before starting HW#6?

Complete SQL Lab #2: Introduction to SQL SELECT. Complete before class on Wednesday (03/19).

Complete SQL Lab #3: Learn how to build a SQL SELECT incrementally. Do in class on Wednesday.



Complete SQL Lab #4: Learn how to use group functions and non-correlated sub-queries. Do in class on Wednesday.

IS475/675

Agenda:
03/19/2025

Answer

- Answer questions about HW#5
- Answer questions about SQL

Learn

- Learn how to write the SELECT statement
  - Structure and order of execution
  - Output
  - Functions
  - Conditional logic
  - Grouping

Structure of the SELECT statement

```
SELECT      [all or distinct]
FROM        (table)
WHERE       (condition)
GROUP BY    (grouping fields)
HAVING      (condition)
ORDER BY    (sort fields)
```

Referred to as the "SELECT LIST"

Most often referred to as a SQL "Query"

When a SELECT statement is executed, the result is referred to as a "result table". It is a memory-based table.

Statement Component	Explanation
SELECT	List which columns should be in the result table. Can use functions, do calculations, create new fields, use conditional logic.
FROM	List which database tables are used to create the result table. This is where you will join tables together based on their foreign keys.
WHERE	Put condition(s) to determine what rows from the database tables should be included in the result table.
GROUP BY	Establish fields used to group/aggregate the data.
HAVING	Put condition(s) to determine which groups should be included in the result table.
ORDER BY	Sort the result table.

How to Learn

SQL Lab Exercise 2: Steps through how to write a simple SELECT statement to access data.

- Shows how to use functions and CASE statements in the SELECT list to build new columns
- Shows how to format columns
- Explains how to sort a result table
- Describes simple conditional logic in the WHERE.

SQL Lab Exercise 3: Steps through a process for creating incrementally a simple SELECT statement.

- The goal is to help you learn how to build a query so that it will require minimal testing to ensure that it works correctly
- The lab also introduces some specialized SELECT options

SQL Lab Exercise 4: Introduces the use of two new parts of the SELECT

- Shows how and when to use GROUP BY
- Explains the group functions available in SQL
- Shows how and when to use HAVING