**IST 471 Exam 1 Review**

Fall 2017

**Lecture 1 – Overview of Data Analytics**

**Analytics** refers to the field of data analysis. It involves discovery of meaningful information (in terms of correlations, trends, patterns, etc.) in data via data queries as well as systematic computational and/or statistical analysis of data. It often involves studying past historical data to discover potential trends, to analyze the effects of certain decisions or events, or to evaluate the performance of a given business scenario.

**Data analytics (DA)** is the science or process of examining raw data to uncover hidden patterns, unknown correlations and other useful information that can be used to make better decisions and planning.

**Spectrum of Data Analytics**

* **Descriptive analytics**: It encompasses the set of techniques that summarizes and describes what has happened in the past. (WHAT HAPPENED?)
  + Descriptive Analytics Tools/Methods
    - Data queries & reporting
    - Descriptive statistics
    - Data visualization (data dashboards, etc.)
    - What-if spreadsheet models
* **Predictive analytics**: It consists of techniques that use models constructed from past data to predict the future or ascertain the impact of one variable on another. (WHAT MIGHT HAPPEN?)
  + Example: Analytics of survey data of past purchase behavior can be used to help predict the market share of a new product.
* **Prescriptive Analytics**: It indicates a best course of action to take (WHAT TO DO to make it happen and prevent it from happening?)
  + Referred to as the "final frontier of analytic capabilities,“prescriptive analytics automatically synthesizes big data, multiple disciplines of mathematical sciences and computational sciences, and business rules, to make predictions and then suggests decision options and actions to take advantage of the predictions.

**Lectures 2 & 3 – Basic and Advanced Subqueries**

Multiple Choice Questions:

1. Single-row operators used with single-row subqueries include:
2. =, >, <, >=, <=, and <>.
3. IN, ALL, ANY
4. All of the above
5. Multi-row operators that can be used with multiple-row subqueries include:
6. =, >, <, >=, <=, and <>.
7. IN, ALL, ANY
8. All of the above

3. What is true about multi-row subqueries?

1. They return multiple columns of data as the result of the subquery.
2. They return single row of data from executing the subquery.
3. They return more than one row of data from executing the subquery.

4. Which of the following is true regarding inserting a subquery in an outer query?

1. A subquery must be a complete query in itself—in other words, it must have at least a SELECT and a FROM clause.
2. A subquery, except one in the FROM clause, can’t have an ORDER BY clause.
3. A subquery must be enclosed in parentheses to separate it from the outer query.
4. If you place a subquery in the outer query’s WHERE or HAVING clause, you can do so only on the right side of the comparison operator.
5. All of the above.

5. A subquery must be placed in the outer query's HAVING clause if:

1. The inner query is used to create a temporary table.
2. The value returned by the inner query is to be compared to grouped data in the outer query.
3. The subquery returns a column alias to the outer query.
4. None of the above. In other words, subqueries can't never be used in the outer query's HAVING clause.

6. A subquery must be placed in the outer query's FROM clause if:

1. The inner query is used to create a temporary table.
2. The value returned by the inner query is to be compared to grouped data in the outer query.
3. The subquery returns a column alias to the outer query.
4. None of the above. In other words, subqueries can't be used in the outer query's FROM clause.

7. A subquery cannot contain a(n) \_\_\_\_ clause except that it is nested in a FROM clause.

1. ORDER BY
2. WHERE
3. HAVING
4. SELECT

8. Which of the following situations that a subquery is suitable?

1. When you need to find all customers living in the state of Florida.
2. When you need to find all publishers who have a telephone number starting with ‘800’.
3. When you need to find all books published by Publisher 4.
4. When you need to find the titles of all books shipped on the same date as an order placed by a particular customer.

9. Which of the following statements is true about subqueries?

1. Subqueries are executed after the outer query is executed.
2. Subqueries and the outer query are executed simultaneously.
3. Subqueries are executed before the outer query is executed.

10. Which of the following is true about the result of a subquery?

1. The result of a subquery is saved in the buffer for future execution but it won’t be used by the outer query.
2. The result of a subquery doesn't give a result. It simply helps boost the speed of the outer query execution.
3. The result of a subquery is used by the outer query.
4. The result of a subquery is always a single value.

11. Which of the following clause is mandatorily used in a subquery? (multiple answers)

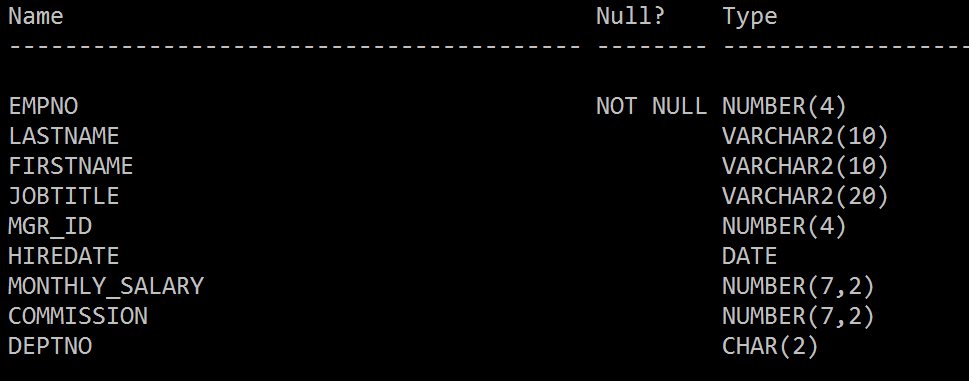
1. SELECT
2. FROM
3. WHERE
4. ORDER BY
5. HAVING

12. In which of the following clauses can a subquery be used?

1. SELECT
2. HAVING
3. WHERE
4. FROM
5. All of the above

**CASE STUDY 1: Data Analysis of EMPLOYEES**

Consider the following **EMP** table:



**1. We are to find the salaries for all the employees who have a higher salary than the IT manager. Which of the following queries will produce the required result?**

a)

SELECT firstname, lastname, monthly\_salary

FROM emp

WHERE monthly\_salary

> (SELECT monthly\_salary FROM emp WHERE jobtitle = 'IT MANAGER');

b)

SELECT firstname, lastname, monthly\_salary

FROM emp

WHERE monthly\_salary = MAX(monthly\_salary)

WHERE jobtitle = 'IT MANAGER';

c)

SELECT firstname, lastname, monthly\_salary

FROM emp WHERE jobtitle= 'IT MANAGER');

**2. Select a proper comparison operator to address the following questions:**

SELECT empno, monthly\_salary

FROM emp

WHERE monthly\_salary =ANY (SELECT monthly\_salary FROM emp WHERE deptno = 10);

**2.1 Find the employees whose salary is equal to the salary of at least one employee in Department 10. (i.e., equal to any of the list)**

1. =
2. IN
3. =ANY
4. >ANY
5. <ANY
6. >ALL
7. <ALL

**2.2 Find out all the employees who have salary greater than at least one employee in Department 10. (i.e., higher than the lowest)**

1. =
2. IN
3. =ANY
4. >ANY
5. <ANY
6. >ALL
7. <ALL

**2.3 Find out all the employees who have salary lesser than the salary of all the employees in Department 10. (i.e., lower than the lowest)**

1. =
2. IN
3. =ANY
4. >ANY
5. <ANY
6. >ALL
7. <ALL

**3. The following two queries will prompt an error message. What’s wrong with them?**

SELECT empno, monthly\_salary

FROM emp

WHERE monthly\_salary =

(SELECT monthly\_salary FROM emp WHERE deptno = 10);

The equals sign produces too many records so it needs to be changed to =ANY

SELECT lastname

FROM emp

WHERE jobtitle =

(SELECT jobtitle FROM emp WHERE empno < 7500);

The equals sign produces too many records so it needs to be changed to =ANY

**4. What will be the outcome of the following query?**

SELECT lastname, jobtitle, monthly\_salary

FROM emp

WHERE monthly\_salary =

(SELECT monthly\_salary FROM emp WHERE jobtitle = 'CLERK')

AND jobtitle <> 'CLERK';

1. It returns those non-clerk employees whose monthly pay is higher than the highest-paid clerk.
2. It returns those non-clerk employees whose monthly pay is lower than the lowest-paid clerk.
3. It returns those non-clerk employees whose monthly pay is less than the highest-paid clerk.
4. It returns those non-clerk employees whose monthly pay is higher than the lowest-paid clerk.
5. It will not execute successfully and prompt an error message “single-row subquery returns more than one row.”

**5. What will be the outcome of the following query?**

SELECT lastname, jobtitle, monthly\_salary

FROM emp

WHERE monthly\_salary >=

(SELECT monthly\_salary FROM emp WHERE jobtitle = 'CLERK')

AND jobtitle <> 'CLERK';

1. It returns those non-clerk employees whose monthly pay is higher than the highest-paid clerk.
2. It returns those non-clerk employees whose monthly pay is lower than the lowest-paid clerk.
3. It returns those non-clerk employees whose monthly pay is less than the highest-paid clerk.
4. It returns those non-clerk employees whose monthly pay is higher than the lowest-paid clerk.
5. It will not execute successfully and prompt an error message “single-row subquery returns more than one row.”

**6. What will be the outcome of the following query?**

SELECT lastname, jobtitle, monthly\_salary

FROM emp

WHERE monthly\_salary >ALL

(SELECT monthly\_salary FROM emp WHERE jobtitle = 'CLERK')

AND jobtitle <> 'CLERK'

ORDER BY monthly\_salary;

1. It returns those non-clerk employees whose monthly pay is higher than the highest-paid clerk.
2. It returns those non-clerk employees whose monthly pay is lower than the lowest-paid clerk.
3. It returns those non-clerk employees whose monthly pay is less than the highest-paid clerk.
4. It returns those non-clerk employees whose monthly pay is higher than the lowest-paid clerk.
5. It will not execute successfully and prompt an error message “single-row subquery returns more than one row.”

**7. What will be the outcome of the following query?**

SELECT lastname, jobtitle, monthly\_salary

FROM emp

WHERE monthly\_salary <ANY

(SELECT monthly\_salary FROM emp WHERE jobtitle = 'CLERK')

AND jobtitle <> 'CLERK'

ORDER BY monthly\_salary;

1. It returns those non-clerk employees whose monthly pay is higher than the highest-paid clerk.
2. It returns those non-clerk employees whose monthly pay is lower than the lowest-paid clerk.
3. It returns those non-clerk employees whose monthly pay is less than the highest-paid clerk.
4. It returns those non-clerk employees whose monthly pay is higher than the lowest-paid clerk.
5. It will not execute successfully and prompt an error message “single-row subquery returns more than one row.”

**8. What will be the outcome of the following query?**

SELECT firstname, lastname, monthly\_salary

FROM emp

WHERE monthly\_salary >

(SELECT MAX(monthly\_salary)

FROM emp

WHERE deptno = 30);

1. It executes successfully and gives the names and salary for those employees whose salary is higher than the highest paid employee in Department 30.
2. It executes successfully and gives the names and salary for those employees whose salary is less than the highest paid employee in Department 30.
3. It throws an error.

**9. The following query will prompt an error. Choose the correct reason for the error as given in the options. (multiple answers)**

SELECT firstname, lastname

FROM emp

WHERE commission =

(SELECT MIN(commission) FROM emp

GROUP BY deptno);

1. The GROUP BY clause is not required in the subquery
2. A function cannot be used in a subquery SELECT statement
3. The single row subquery gives multiple records
4. The use of "=" operator is invalid; an IN operator will work correctly

**10. What will be the outcome of the following query?**

SELECT empno, firstname, lastname, jobtitle, monthly\_salary

FROM emp

WHERE monthly\_salary < ALL

(SELECT monthly\_salary

FROM emp

WHERE jobtitle = 'SALESMAN')

AND jobtitle <> 'SALESMAN';

1. Return the employees who are not a sales person and have a salary less than that of the salary for all of the salespersons.
2. Return the employees who are not a sales person and have a salary greater than that of the salary for all of the salespersons.
3. Return the employees who are not a sales person and have a salary greater than that of the highest paid salesperson.

**11. What will be the outcome of the following query?**

SELECT lastname, jobtitle, monthly\_salary

FROM emp

WHERE monthly\_salary \_\_\_\_\_\_

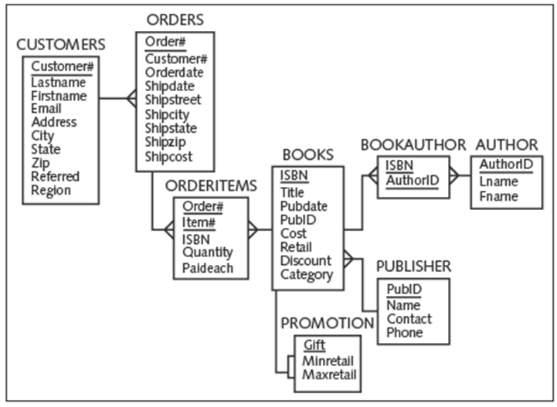
**(SELECT monthly\_salary FROM emp WHERE jobtitle = 'SALESMAN')**

ORDER BY monthly\_salary;

Given that four values are returned by the subquery: 2750, 3000, 3100, and 3750.

|  |  |  |
| --- | --- | --- |
|  | Function | Result |
| >ALL | the outer query searches for all records with a value higher than the highest value returned by the subquery |  |
| <ALL | the outer query searches for all records with a value lower than the lowest value returned by the subquery |  |
| >ANY | the outer query searches for records with a value higher than the lowest value returned by a subquery |  |
| <ANY | the outer query searches for records with a value less than the highest value returned by a subquery |  |
| =ANY | the outer query searches for the records with a value equal to any of the values returned by a subquery |  |
| IN | Same as =ALL |  |

**CASE STUDY 2: JustLee Books**



**1. What will be the outcome of the following query?**

SELECT category, title, cost

FROM books

WHERE cost <

(SELECT cost FROM books

WHERE title = 'DATABASE IMPLEMENTATION')

AND category = 'COMPUTER'

ORDER BY cost;

It selects the book that costs less than DB implementation and gives the category, title and cost of that book

**2. What will be the outcome of the following query?**

SELECT title, retail

FROM books

WHERE cost =

(SELECT MAX(cost) FROM books);

It finds the book that costs the most and outputs the name of that book along with the price of the book

1. **We are to find out the most expensive book. Which of the following queries will give you the required results?**
2. SELECT title, MAX(retail) FROM books;
3. SELECT title, MAX(retail) FROM books GROUP BY title;
4. SELECT title, retail FROM books WHERE retail = (SELECT MAX(retail) FROM books);
5. SELECT title, retail FROM books WHERE retail = MAX(retail);
6. **Which of the following is syntactically correct that involves a subquery?**
7. SELECT order# FROM orders WHERE shipdate = SELECT shipdate FROM orders WHERE order# = 1010;
8. SELECT order# FROM orders WHERE customer# = (SELECT shipdate FROM orders WHERE order# = 1010);
9. SELECT order# FROM orders WHERE shipdate = (SELECT shipdate FROM orders WHERE order# = 1010);
10. **Which of the following statements is valid?**
11. SELECT title FROM books WHERE cost < (SELECT cost FROM books WHERE isbn = '9959789321');
12. SELECT title FROM books WHERE cost = (SELECT cost FROM books WHERE isbn = '9959789321' ORDER BY cost);
13. SELECT title FROM books WHERE category = (SELECT cost FROM orderitems HAVING isbn = '9959789321');

**6. Which of the following queries is a valid query to identify those customers who live in the same state as the customer named Leila Smith?**

1. SELECT customer# FROM customers WHERE state =

(SELECT state FROM customers WHERE lastname = 'SMITH' AND firstname = 'LEILA');

1. SELECT customer# FROM customers WHERE state =

(SELECT customer# FROM customers WHERE lastname = 'SMITH' AND firstname = 'LEILA');

**7. Study the following SQL statement:**

SELECT customer# FROM customers WHERE referred =

(SELECT referred FROM customers WHERE firstname = 'JORGE' AND lastname = 'PEREZ');

What is the purpose of the SQL statement listed above?

1. To identify other customers who were referred to JustLee Books by the same person who referred Jorge Perez.
2. To identify other customers who were referred to JustLee Books by the same person who was referred by Jorge Perez.

8. **We are to find out those books with a retail price greater than the average retail price of other books in the same category.**

a)

SELECT title, retail, category, retail FROM books

WHERE retail = (SELECT AVG(retail) FROM books GROUP BY category);

b)

SELECT b.title, b.retail, t.category, t.cataverage

FROM books b,

(SELECT category, AVG(retail) cataverage FROM books GROUP BY category) t

WHERE b.category = t.category;

**9. Exam the following query:**

SELECT b.title, b.cost, t.category, t.catalog\_max

FROM books b,

(SELECT category, MAX(cost) catalog\_max FROM books GROUP BY category) t

WHERE b.category = t.category;

The inner query above is considered a \_\_\_\_\_ subquery. The output can be used as a temporary table that stores data of the average retail price for each category.

1. Single-column, single-row
2. single-column, multiple-row
3. multiple-column