

Version: 03.01

Date: 09-Mar-2016

Developed by:

Verified by:

Endava SQL Discipline

SQL Test

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date of revision** | **Description of modifications** |
| 03.01 | 11-Mar-2016 | The third version of the document. |

# Description

**About the test:**

|  |  |
| --- | --- |
|  |  |
| Applied Level | Intermediate |
| Number of tasks | 13 |
| Domain | Standard DB |
| Test type | PC |
| Test duration | 2 h |

**Evaluation info:**

|  |  |
| --- | --- |
|  |  |
| Evaluated person |  |
| Evaluator name |  |
| Date of evaluation |  |
| Evaluation result  (passed/failed) |  |

# Test Tasks

## Precondition:

Use the below credentials:

|  |  |
| --- | --- |
| **Server Name** | MDCH-AMWCI-S01 |
|
| **DB** | AMInternship |
|
|

## Task 1

**What is JOIN and explain different types of joins that you know.**

**Answer:**

|  |
| --- |
| A JOIN clause is used to combine rows from two or more tables, based on a related column between them. Different Types of SQL JOINs Here are the different types of the JOINs in SQL:   * **(INNER) JOIN**: Returns records that have matching values in both tables * **LEFT (OUTER) JOIN**: Return all records from the left table, and the matched records from the right table * **RIGHT (OUTER) JOIN**: Return all records from the right table, and the matched records from the left table * **FULL (OUTER) JOIN**: Return all records when there is a match in either left or right table |

## Task 2

**What is the difference between UNION and UNION ALL?**

**Answer:**

|  |
| --- |
| The UNION operator is used to combine the result-set of two or more SELECT statements.  The **UNION** operator selects only **distinct** values by default. To allow duplicate values, use **UNION ALL** |

## Task 3

**What is a Cartesian product?**

**Answer:**

|  |
| --- |
| The **CARTESIAN JOIN** or **CROSS JOIN** returns the Cartesian product of the sets of records from the two or more joined tables. Thus, it equates to an inner join where the join-condition always evaluates to True or where the join-condition is absent from the statement. |

## Task 4

**Display all the streets and street numbers from PANAMA country.**

**Query:**

|  |
| --- |
| /\*1st method\*/  select address.street,address.street\_num  from address where address.city\_id in  ( select id from city where city.country\_id in  (select id from country where name='Panama')  )  /\*2nd method\*/  select a.street, a.street\_num  from address a  inner join country c ON a.city\_id=c.id  where c.name='Panama' |

## Task 5

**Display the street names and all the persons who live in PANAMA country and have street number bigger than 100.**

**Query:**

|  |
| --- |
| /\*1st method\*/  SELECT p.first\_name,p.last\_name, a.street  FROM person p, address a  WHERE p.address\_id IN (  SELECT a.id from address WHERE a.city\_id IN  (  SELECT city.id from city WHERE city.country\_id IN  (  SELECT country.id FROM country WHERE country.name = 'Panama'  )  )  )  AND a.street\_num > 100  /\*2nd method\*/  select p.first\_name, p.last\_name, a.street, a.street\_num  from address a  inner join country c ON a.city\_id=c.id  inner join person p ON p.address\_id=a.id  where c.name='Panama' AND a.street\_num>100 |

## Task 6

**Display all the details of the persons who are working in the “Blogtag” company and have the salary bigger than 2000 EUR.**

**Query:**

|  |
| --- |
| /\*1st method\*/  SELECT p.first\_name,p.last\_name  FROM person p  WHERE p.job\_id IN (  SELECT id from job WHERE company\_id IN ( SELECT id from company WHERE name='Blogtag' )  AND  job.salary\_id in (select id from salary where salary>2000 )    )  /\*2nd method\*/  select p.first\_name, p.last\_name, p.date,c.name,s.salary,s.currency  from job j  inner join person p ON p.job\_id=j.id  inner join company c ON c.id=j.company\_id  inner join salary s ON s.id=j.salary\_id  where s.salary>2000 AND s.currency='EUR' AND c.name='Blogtag' |

## Task 7

**Display all the person’s details and the city where they are from which are working in the “Blogtag” company and have the salary bigger than 2000 EUR.**

**Query:**

|  |
| --- |
| select p.first\_name, p.last\_name, p.date,c.name,s.salary,s.currency, cit.name  from person p  inner join job j ON p.job\_id=j.id  inner join company c ON c.id=j.company\_id  inner join salary s ON s.id=j.salary\_id  inner join address a ON p.address\_id=a.id  inner join city cit ON cit.id=a.city\_id  where s.salary>2000 AND s.currency='EUR' AND c.name='Blogtag' |

## Task 8

**Display the person’s details and the city where they are from which are male and working in the “Blogtag” company and have the salary bigger than 2000 EUR.**

**Query:**

|  |
| --- |
| select p.first\_name, p.last\_name, p.date,c.name,s.salary,s.currency, cit.name  from person p  inner join job j ON p.job\_id=j.id  inner join company c ON c.id=j.company\_id  inner join salary s ON s.id=j.salary\_id  inner join address a ON p.address\_id=a.id  inner join gender g ON g.id=p.gender\_id  inner join city cit ON cit.id=a.city\_id  where s.salary>2000 AND s.currency='EUR' AND c.name='Blogtag'  And g.name='male' |

## Task 9

**Display all the persons who have the username starting with “b” and have the ip address starting with “14”.**

**Query:**

|  |
| --- |
| select p.first\_name, p.last\_name, w.ip\_address, w.username  from job j  inner join person p ON p.job\_id=j.id  inner join workplace w ON w.id=j.workplace\_id  where w.username like 'b%'  AND  w.ip\_address like '14%' |

## Task 10

**Display which ladies have the username starting with “b” and the job title contains “Environmental”.**

**Query:**

|  |
| --- |
| select p.first\_name, p.last\_name, w.ip\_address, w.username  from job j  inner join person p ON p.job\_id=j.id  inner join workplace w ON w.id=j.workplace\_id  inner join gender g ON p.gender\_id=g.id  inner join job\_title jt ON jt.id=j.jobtitle\_id  where w.username like 'b%'  AND  w.ip\_address like '14%'  And g.name='female' AND jt.title like '%Environmental%' |

## Task 11

**Insert a new record in the Person table (on your database) with the information: first name – Camelia, last name – Agachi, gender, address „Kingsford 8” with the job titile „Environmental Tech” and notice the time of result set.**

**Query:**

|  |
| --- |
| insert into person values ('Camelia','Agachi','1988-01-02',2,6,16)  00:00:00.031 |

## Task 12

**Create 2 tables, for example A and B with 2 columns each: Number (int) and Text\_number (varchar).**

**Insert 15 records for each table:**

1. **First table should contain the values from 1 to 15 (i.e. (1, ‘one’) )**
2. **The second table should contain the values from 11 to 25 (i.e. (11, ‘eleven’) )**

**Query:**

|  |
| --- |
| create table A (  number int ,  text\_number varchar(50)  )  create table B (  number int ,  text\_number varchar(50)  )  insert into A values (1,'one')  insert into A values (2,'two')  insert into A values (3,'three')  insert into A values (4,'four')  insert into A values (5,'five')  insert into A values (6,'six')  insert into A values (7,'seven')  insert into A values (8,'eight')  insert into A values (9,'nine')  insert into A values (10,'ten')  insert into A values (11,'eleven')  insert into A values (12,'twelve')  insert into A values (13,'thirteen')  insert into A values (14,'fourteen')  insert into A values (15,'fifteen')  insert into B values (11,'eleven')  insert into B values (12,'twelve')  insert into B values (13,'thirteen')  insert into B values (14,'fourteen')  insert into B values (15,'fifteen')  insert into B values (16,'sixteen')  insert into B values (17,'seventeen')  insert into B values (18,'eighteen')  insert into B values (19,'nineteen')  insert into B values (20,'twenty')  insert into B values (21,'twenty-one')  insert into B values (22,'twenty-two')  insert into B values (23,'twenty-three')  insert into B values (24,'twenty-four')  insert into B values (25,'twenty-five') |

## Task 13

**Use INNER JOIN, LEFT, RIGHT, FULL OUTER JOINs and UNION and UNION ALL on the above 2 tables and present the results.**

**Query:**

|  |
| --- |
| select \* from B  inner join A ON a.number=B.number    select \* from B  left join A ON b.number=a.number    select \* from A  left join B ON b.number=a.number    select \* from A  right join B ON b.number=a.number    select \* from B  right join A ON b.number=a.number    select \* from A  Full outer join B ON b.number=a.number    select \* from A  union  Select \* from B    select \* from A  union ALL  Select \* from B  1 one  2 two  3 three  4 four  5 five  6 six  7 seven  8 eight  9 nine  10 ten  11 eleven  12 twelve  13 thirteen  14 fourteen  15 fifteen  11 eleven  12 twelve  13 thirteen  14 fourteen  15 fifteen  16 sixteen  17 seventeen  18 eighteen  19 nineteen  20 twenty  21 twenty-one  22 twenty-two  23 twenty-three  24 twenty-four  25 twenty-five |