

Version: 03.01

Date: 10-Mar-2016

Developed by:

Verified by:

Endava SQL Discipline

SQL Test

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date of revision** | **Description of modifications** |
| 03.01 | 10-MAR-2016 | The third version of the document. |

# Description

**About the test:**

|  |  |
| --- | --- |
|  |  |
| Applied Level | Basic |
| Number of tasks | 8 |
| 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

**Which are the two main components of SQL language? Importance of these two components.**

**Answer:**

|  |
| --- |
| SQL consists of three components:   1. Data Definition Language (DDL) 2. Data Manipulation Language (DML)   **The Data Definition Language (DDL).**This component of the SQL language is used to create and modify tables and other objects in the database. For tables there are three main commands:  CREATE TABLE tablename to create a table in the database DROP TABLE tablename to remove a table from the database ALTER TABLE tablename to add or remove columns from a table in the database  **The Data Manipulation Language (DML)**component of the SQL language is used to manipulate data within a table. There are four main commands:  SELECT to select rows of data from a table INSERT to insert rows of data into a table UPDATE to change rows of data in a table DELETE to remove rows of data from a table |

## Task 2

**Which are the advantages and disadvantages of SQL language?**

**Answer:**

|  |
| --- |
| **Advantages of SQL:**  \* High Speed:  SQL Queries can be used to retrieve large amounts of records from a database quickly and efficiently.  \* Well Defined Standards Exist:  SQL databases use long-established standard, which is being adopted by ANSI & ISO. Non-SQL  databases do not adhere to any clear standard.  \* No Coding Required: Using standard SQL it is easier to manage database systems without having to write substantial amount of code.  \* Emergence of ORDBMS:  Previously SQL databases were synonymous with relational database. With the emergence of Object  Oriented DBMS, object storage capabilities are extended to relational databases.  **Disadvantages of SQL:**  \* Difficulty in Interfacing: Interfacing an SQL database is more complex than adding a few lines of code.  \* More Features Implemented in Proprietary way: Although SQL databases conform to ANSI & ISO  standards, some databases go for proprietary extensions to standard SQL to ensure vendor lock-in |

## Task 3

**Which are the different contraints and their meanings in SQL Server?**

**Answer:**

|  |
| --- |
| The following constraints are commonly used in SQL:   * [**NOT NULL**](https://www.w3schools.com/sql/sql_notnull.asp) - Ensures that a column cannot have a NULL value * [**UNIQUE**](https://www.w3schools.com/sql/sql_unique.asp) - Ensures that all values in a column are different * [**PRIMARY KEY**](https://www.w3schools.com/sql/sql_primarykey.asp) - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table * [**FOREIGN KEY**](https://www.w3schools.com/sql/sql_foreignkey.asp) - Uniquely identifies a row/record in another table * [**CHECK**](https://www.w3schools.com/sql/sql_check.asp) - Ensures that all values in a column satisfies a specific condition * [**DEFAULT**](https://www.w3schools.com/sql/sql_default.asp) - Sets a default value for a column when no value is specified * [**INDEX**](https://www.w3schools.com/sql/sql_create_index.asp) - Use to create and retrieve data from the database very quickly |

## Task 4

**What's the difference between the Primary Key and Unique Key?**

**Answer:**

|  |
| --- |
| **Primary Key:**   * There an only be one primary key in a table * In some DBMS it cannot be NULL - e.g. MySQL adds NOT NULL * Primary Key is a unique key identifier of the record   **Unique Key:**   * Can be more than one unique key in one table * Unique key can have null values * It can be a candidate key * Unique key can be null and may not be unique |

## Task 5

**What is the difference between DELETE, DROP and TRUNCATE statements?**

**Answer:**

|  |
| --- |
| DELETE The DELETE command is used to remove rows from a table. A WHERE clause can be used to only remove some rows. If no WHERE condition is specified, all rows will be removed. After performing a DELETE operation you need to COMMIT or ROLLBACK the transaction to make the change permanent or to undo it. Note that this operation will cause all DELETE triggers on the table to fire. TRUNCATE TRUNCATE removes **all rows** from a table. The operation cannot be rolled back and no triggers will be fired. As such, TRUCATE is faster and doesn't use as much undo space as a DELETE. DROP The DROP command removes a table from the database. All the tables' rows, indexes and privileges will also be removed. No DML triggers will be fired. The operation cannot be rolled back.  DROP and TRUNCATE are DDL commands, whereas DELETE is a DML command. Therefore DELETE operations can be rolled back (undone), while DROP and TRUNCATE operations cannot be rolled back. |

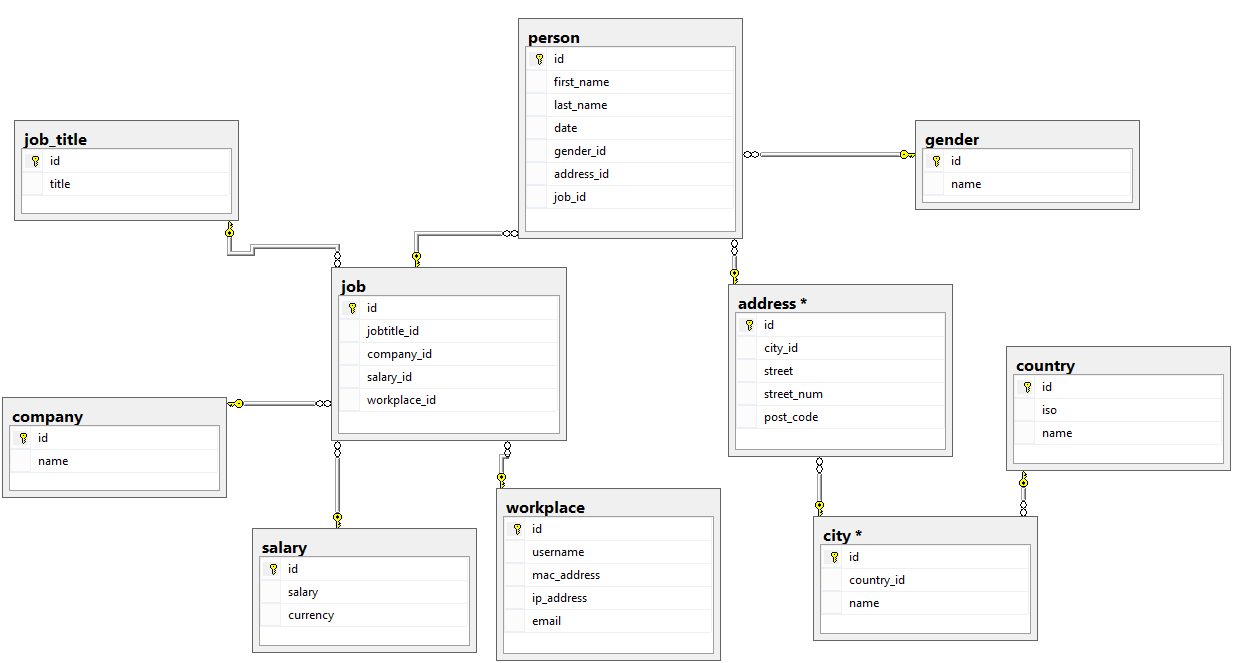
## Task 6

**Replicate the database AMInternship using the name AMInternship\_YourNameSurname by recreating the existing tables.**

**Query:**

|  |
| --- |
| /\*create table address\*/  create table address (  id int NOT NULL Primary key ,  city\_id int NOT NULL,  street varchar(50),  street\_num varchar(50),  post\_code varchar(50)  )  /\*create table city\*/  create table city (  id int NOT NULL Primary key ,  country\_id int NOT NULL,  name varchar(50),  )  /\*create table company\*/  create table company (  id int NOT NULL Primary key ,  name varchar(50)  )  /\*create table country\*/  create table country (  id int NOT NULL Primary key ,  iso varchar(2) NOT NULL,  name varchar(50) NOT NULL,  )  /\*create table gender\*/  create table gender (  id int NOT NULL Primary key ,  name varchar(50) NOT NULL,  )  /\*create table job\*/  create table job (  id int NOT NULL Primary key ,  jobtitle\_id int NOT NULL,  company\_id int NOT NULL,  salary\_id int NOT NULL,  workplace\_id int NOT NULL  )  /\*create table job\_title\*/  create table job\_title (  id int NOT NULL Primary key ,  title varchar(50) NOT NULL,  )  /\*create table person\*/  create table person (  id int NOT NULL Primary key ,  first\_name varchar(50) NOT NULL,  last\_name varchar(50) NOT NULL,  date date NOT NULL,  gender\_id int NOT NULL,  address\_id int NOT NULL,  job\_id int NOT NULL,  )  /\*create table salary\*/  create table salary (  id int NOT NULL Primary key ,  salary int NOT NULL ,  currency varchar(10) NOT NULL,  )  /\*create table workplace\*/  create table workplace (  id int NOT NULL Primary key ,  username varchar(50) NOT NULL,  mac\_address varchar(50) NOT NULL,  ip\_address varchar(50) NOT NULL,  email varchar(50) NOT NULL,  ) |

## Task 7

****

**Add the PRIMARY and FOREIGN KEY constraints for the created tables accordingly.**

**\*You can also add the NOT NULL and UNIQUE constraints if you consider that it’s needed.**

**Query:**

|  |
| --- |
| ALTER TABLE job  ADD FOREIGN KEY (company\_id) REFERENCES company(id);  ALTER TABLE job  ADD FOREIGN KEY (jobtitle\_id) REFERENCES job\_title(id);  ALTER TABLE job  ADD FOREIGN KEY (salary\_id) REFERENCES salary(id);  ALTER TABLE job  ADD FOREIGN KEY (workplace\_id) REFERENCES workplace(id);  ALTER TABLE person  ADD FOREIGN KEY (job\_id) REFERENCES job(id);  ALTER TABLE person  ADD FOREIGN KEY (gender\_id) REFERENCES gender(id);  ALTER TABLE person  ADD FOREIGN KEY (address\_id) REFERENCES address(id);  ALTER TABLE address  ADD FOREIGN KEY (city\_id) REFERENCES city(id);  ALTER TABLE city  ADD FOREIGN KEY (country\_id) REFERENCES country(id); |

## Task 8

**Add the DEFAULT constraint on Street column with the default value „Endava” for the table [Address].**

**Query:**

|  |
| --- |
| ALTER TABLE address  ALTER COLUMN Street SET DEFAULT 'Endava'; |