**Project : 40364**

For Learner Use:

|  |  |
| --- | --- |
| Surname of Learner | Naidu |
| Name of Learner | Kian |
| Learner ID | 0110105463084 |
| Student Number | KNPMB070 |
| Date of Project Given | 08/06/2020 |
| Location / Branch | Durban |

For Assessors Use:

|  |  |
| --- | --- |
| Name of Branch | Durban |
| Name of Facilitator |  |
| Name of Assessor |  |
| Date Marked / Assessed |  |
| Mark Allocation | 55 |
| Mark Obtained |  |
| Competency Status ( C / NYC) |  |

|  |  |
| --- | --- |
| Unit Standards |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Candidates Signature |  | Date Submitted | 21/06/2020 |
| Assessors Signature |  | Date  Marked |  |

With the use of SQL Server notes and Database Fundamentals textbook, do the following tasks.

1. Write a query to create a database called SALES. **[1]**

|  |
| --- |
| CREATE DATABASE SALES |

1. Write a T-SQL statement to create the table called CLIENT as follows. **[4]**

CLIENT\_ID NUMERIC [1, 0]

FIRSTNAME NVARCHAR [20]

LASTNAME NVARCHAR [20]

SALESDATE DATE

AGE INT

|  |
| --- |
| USE SALES;  CREATE TABLE CLIENT  (  [CLIENT\_ID] numeric(1,0),  [FIRSTNAME] nvarchar(20),  [LASTNAME] nvarchar(20),  [SALESDATE] date,  [AGE] int  ); |

1. Write a query to add the following data into CLIENT table.`  **[6]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLIENT\_ID** | **FIRSTNAME** | **LASTNAME** | **SALESDATE** | **AGE** |
| 1 | Fred | Smith | 12-JUL-1990 | 55 |
| 2 | Errol | Ngidi | 30-MAR-1973 | 27 |
| 3 | Faith | Brown | 10-DEC-1986 | 33 |
| 4 | Linda | Mhlongo | 04-OCT-1987 | 42 |
| 5 | Ronald | Naidoo | 22-SEPT-1992 | 25 |
| 6 | Owen | Zungu | 17-APR-1989 | 19 |

|  |
| --- |
| USE SALES;  INSERT INTO CLIENT VALUES  (1,'Fred','Smith', '1990-07-12',55),  (2,'Errol','Ngidi', '1973-03-30',27),  (3,'Faith','Brown', '1986-12-10',33),  (4,'Linda','Mhlongo', '1987-10-04',42),  (5,'Ronald','Naidoo', '1992-09-22',25),  (6,'Owen','Zungu', '1989-04-17',19); |

1. Write a T-SQL statement to create the table called PRODUCT as follows. **[4]**

PRODUCT\_ID NUMERIC [3, 0]

PROD\_NAME NVARCHAR [20]

PRICE MONEY

SHIPMENT\_DATE DATE

QUANTITY INT

|  |
| --- |
| USE SALES;  CREATE TABLE PRODUCT  (  [PRODUCT\_ID] NUMERIC (3, 0),  [PROD\_NAME] NVARCHAR (20),  [PRICE] MONEY,  [SHIPMENT\_DATE] DATE,  [QUANTITY] INT  ); |

1. Write a query to add the following data into CLIENT table. **[6]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROD\_ID** | **PROD\_NAME** | **PRICE** | **SHIPMENT\_DATE** | **QUANTITY** |
| 101 | LETTICE | 40.00 | 22-SEPT-2011 | 10 |
| 102 | JAM TOMATO | 56.33 | 17-APR-2010 | 7 |
| 103 | RED PAPPER | 24.00 | 12-JUL-2012 | 31 |
| 104 | CARROT | 100.00 | 30-MAR-1973 |  |
| 105 | LETTICE | 79.00 | 10-DEC-2013 | 22 |
| 106 | CARROT | 24.00 | 04-OCT-2012 | 31 |

|  |
| --- |
| USE SALES;  INSERT INTO PRODUCT VALUES  (101,'LETTICE',40,'2011-09-22',10),  (102,'JAM TOMATO',56.33,'2010-04-17',7),  (103,'RED PAPPER',24,'2012-07-12',31),  (104,'CARROT',100,'1973-03-30',null),  (105,'LETTICE',79,'2013-12-10',22),  (106,'CARROT',24,'2012-10-04',31); |

1. Use the DML and DDL languages to create the following statements; **[8]**
2. View all the records of a CLIENT table. [1]

|  |
| --- |
| SELECT \* FROM CLIENT; |

1. View the record with the product\_id of 104. [2]

|  |
| --- |
| SELECT \* FROM PRODUCT WHERE PRODUCT\_ID = 104; |

1. View all the products that cost more than R50.00. Hint: WHERE clause, operators such as >, <, +, =,!= [2]

|  |
| --- |
| SELECT PROD\_NAME FROM PRODUCT WHERE PRICE >50; |

1. Delete all the records where the shipment date is before 01-JAN-2012. Hint: WHERE clause, operators such as >, <, +, =,!= [2]

|  |
| --- |
| DELETE FROM PRODUCT WHERE SHIPMENT\_DATE < '2012-01-01'; |

1. Write a query that will count all rows from CLIENT table. [1]

|  |
| --- |
| SELECT COUNT(\*) AS [TotalRows] FROM CLIENT; |

1. Explain what a primary key is.  **[1]**

|  |
| --- |
| A column (or combination of columns) used to uniquely identify a record. |

1. Explain 5 characteristics of a primary key that are considered desirable. **[5]**

|  |
| --- |
| 1. Must contain a unique value for each row of data it has  2. No change over time, the value is independent and is not prone to changes  3. Nonintelligent, the value is meaningless  4. Preferably a numeric value, so a value can’t easily be repeated and there is a sequential order.  5. Preferably single attribute, the primary key should have the minimum number of attributes as possible to make it easier to work with. |

1. Use Lesson 04 to assist you with Answer the following questions **[10]**
2. Explain what Normalization is. [2]
3. What are two conditions to be met for a table to be in 2NF (Second Normal Form)? [2]
4. List three main objectives of normalizing tables. [2]
5. What is a composite key? [2]
6. Explain what is meant by redundant data. [2]

|  |
| --- |
| a) The process of eliminating redundant data to save space.  b)   * Needs to be in the first normal form. * The attributes of an entity need to be fully dependant on the entire primary key.   c)   * To use storage space efficiently. * The elimination of redundant data. * To reduce and possibly eliminate inconsistent data.   d) A combination of two or more attributes to uniquely identify a each row.  e) Data saved in multiple locations that gives rise to the same information which is being repeated unnecessarily. |

1. **In your own words** define the following terms. No copy and paste. **[10]**
2. Stored procedure. [2]
3. Data type. [2]
4. View. [2]
5. Difference between VARCHAR and NVARCHAR [2]
6. Relational database [2]

|  |
| --- |
| a) SQL statements saved in a database which makes certain SQL statements easier to access and reuse multiple times.  b) A set of values data can be taken in and stored as with different formats and sizes depending on the needed specifications.  c) A virtual table that is based on a predefined SQL statement which gets data from one or more table columns.  d) nvarchar supports English and non-english symbols and takes up two bytes of space whereas varchar supports English only characters and is only one byte in size.  e) A database that has tables linked in relationships that have a correlation with one another in a one to many or many to many relationship that allow for easy reference to linked data. |

*\*\*END OF ASSESSMENT\*\**