CS2005-Database Systems

Assignment 1

Faizan Shabir

22L-6552

4H

**Part 1**

**Datatypes:**

MySQL

* DATE - format YYYY-MM-DD
* DATETIME - format: YYYY-MM-DD HH:MI:SS
* TIMESTAMP - format: YYYY-MM-DD HH:MI:SS
* YEAR - format YYYY or YY

Oracle

1. DATE:
   * Valid date range: January 1, 4712 BC, to December 31, 9999 AD.
   * Size: Fixed at 7 bytes.
   * Contains datetime fields: YEAR, MONTH, DAY, HOUR, MINUTE, and SECOND.
   * Default format determined by NLS\_DATE\_FORMAT or NLS\_TERRITORY.
   * No fractional seconds or time zone.
2. TIMESTAMP [(fractional\_seconds\_precision)]:
   * Combines date and time with fractional seconds.
   * Size: 7 or 11 bytes (depending on precision).
   * Contains datetime fields: YEAR, MONTH, DAY, HOUR, MINUTE, and SECOND.
   * Default format determined by NLS\_TIMESTAMP\_FORMAT or NLS\_TERRITORY.
   * No time zone information.
3. TIMESTAMP [(fractional\_seconds\_precision)] WITH TIME ZONE:
   * Includes TIMESTAMP values and time zone displacement.
   * Size: Fixed at 13 bytes.
   * Contains datetime fields: YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, TIMEZONE\_HOUR, and TIMEZONE\_MINUTE.
   * Has fractional seconds and explicit time zone.
4. TIMESTAMP [(fractional\_seconds\_precision)] WITH LOCAL TIME ZONE:
   * Similar to TIMESTAMP WITH TIME ZONE but normalized to the database time zone when stored.
   * Size: 7 or 11 bytes (depending on precision).
   * Data retrieved in the session time zone.
   * Default format determined by NLS\_TIMESTAMP\_FORMAT or NLS\_TERRITORY.
5. INTERVAL YEAR [(year\_precision)] TO MONTH:
   * Stores a period in years and months.
   * Size: Fixed at 5 bytes.
   * Year precision determines the number of digits in the YEAR field.
6. INTERVAL DAY [(day\_precision)] TO SECOND [(fractional\_seconds\_precision)]:
   * Stores a period in days, hours, minutes, and seconds.
   * Day precision sets the maximum digits in the DAY field.
   * Fractional seconds precision determines digits in the SECOND field.

SQL Server

* DATETIME: From January 1, 1753 to December 31, 9999 with an accuracy of 3.33 milliseconds (8 bytes).
* Datetime2: From January 1, 0001 to December 31, 9999 with an accuracy of 100 nanoseconds (6-8 bytes)
* Smalldatetime: From January 1, 1900 to June 6, 2079 with an accuracy of 1 minute (4 bytes)
* Date: Store a date only. From January 1, 0001 to December 31, 9999 (3 bytes)
* Time: Store a time only to an accuracy of 100 nanoseconds (3-5 bytes)
* Datetimeoffset: The same as datetime2 with the addition of a time zone offset (8-10 bytes)
* Timestamp: Stores a unique number that gets updated every time a row gets created or modified. The timestamp value is based upon an internal clock and does not correspond to real time. Each table may have only one timestamp variable

**Date Functions and usage in queries:**

MySQL

* **date():**
  + Get the date from a given date/datetime.
* **adddata():**
  + Get the date with added time/date intervals.
* **curdate():**
  + Get the current date.
* **current\_date():**
  + Get the current date.
* **date\_add():**
  + Get the date with added date/datetime intervals.
* **date\_format():**
  + Get the date in a specified format.
* **datediff():**
  + Get the difference between two specified date values.
* **day():**
  + Get the day from a given date.
* **dayname():**
  + Get the name of the day from a given date.
* **dayofmonth():**
  + Get the day for a specified date.
* **dayofweek():**
  + Get the day of the week in numeric.
* **dayofyear():**
  + Get the number of days in the year.
* **from\_days():**
  + Get the date from a given number of days.
* **hour():**
  + Get the hour from a given datetime.
* **addtime():**
  + Get the time/datetime with added time intervals.
* **current\_time():**
  + Get the current time.
* **current\_timestamp():**
  + Get the current date and time.
* **curtime():**
  + Get the current time.
* **last\_day():**
  + Get the last date of a given month.
* **localtime() / localtimestamp():**
  + Get the current date and time.
* **makedate():**
  + Make the date from a given year and number of days.
* **maketime():**
  + Make the time from given hour, minute, and second.
* **microsecond():**
  + Get the value of microsecond from a given datetime or time.
* **minute():**
  + Get the value of the minute for a specified datetime or time.
* **month():**
  + Get the value of the month from a given datetime or time.
* **monthname():**
  + Get the full month name.
* **now():**
  + Get the current date and time.
* **period\_add():**
  + Add a given number of months to a period.
* **period\_diff():**
  + Get the difference between two periods.
* **quarter():**
  + Get the quarter portion of a specified date/datetime.
* **sec\_to\_time():**
  + Convert specified seconds into time.
* **second():**
  + Get the second portion from a specified date/datetime.
* **str\_to\_date():**
  + Convert a string into the given format mask.
* **subdate():**
  + Get the date subtracted by given intervals.
* **subtime():**
  + Get the time/datetime subtracted by certain intervals.
* **sysdate():**
  + Get the system date.
* **time():**
  + Get the time for the given time/datetime.
* **time\_format():**
  + Format the time in a specified format mask.
* **time\_to\_sec():**
  + Convert the time into seconds.
* **timediff():**
  + Get the difference for the given two time/datetime.
* **timestamp():**
  + Convert the expression into datetime/time.
* **to\_day():**
  + Convert the date into the numeric number of days.
* **weekday():**
  + Get the index for a date.
* **week():**
  + Get the week portion for the specified date.
* **weekofyear():**
  + Get the week of the given date.

Top of Form

Example 1: SELECT ADDDATE('1997-01-02', 31);

Example 2: SELECT DATE('2004-12-21 01:02:03');

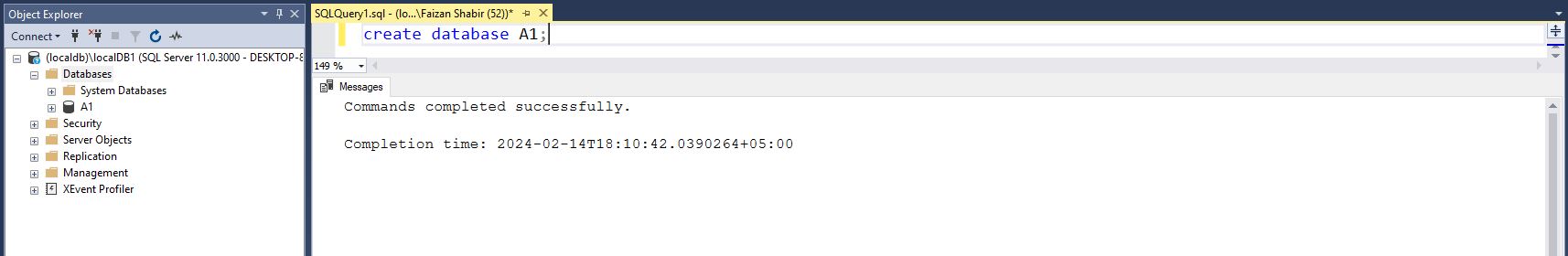
Oracle

| **Function** | **Example** | **Description** |
| --- | --- | --- |
| [ADD\_MONTHS](https://www.oracletutorial.com/oracle-date-functions/oracle-add_months/) | ADD\_MONTHS( DATE '2016-02-29', 1 ) | Add a number of months (n) to a date and return the same day which is n of months away. | |
| [CURRENT\_DATE](https://www.oracletutorial.com/oracle-date-functions/oracle-current_date/) | SELECT CURRENT\_DATE FROM dual | Return the current date and time in the session time zone | |
| [CURRENT\_TIMESTAMP](https://www.oracletutorial.com/oracle-date-functions/oracle-current_timestamp/) | SELECT CURRENT\_TIMESTAMP FROM dual | Return the current date and time with time zone in the session time zone | |
| [DBTIMEZONE](https://www.oracletutorial.com/oracle-date-functions/oracle-dbtimezone/) | SELECT DBTIMEZONE FROM dual; | Get the current database time zone | |
| [EXTRACT](https://www.oracletutorial.com/oracle-date-functions/oracle-extract/) | EXTRACT(YEAR FROM SYSDATE) | Extract a value of a date time field e.g., YEAR, MONTH, DAY, … from a date time value. | |
| [FROM\_TZ](https://www.oracletutorial.com/oracle-date-functions/oracle-from_tz/) | FROM\_TZ(TIMESTAMP '2017-08-08 08:09:10', '-09:00') | Convert a timestamp and a time zone to a TIMESTAMP WITH TIME ZONE value | |
| [LAST\_DAY](https://www.oracletutorial.com/oracle-date-functions/oracle-last_day/) | LAST\_DAY(DATE '2016-02-01') | Gets the last day of the month of a specified date. | |
| [LOCALTIMESTAMP](https://www.oracletutorial.com/oracle-date-functions/oracle-localtimestamp/) | SELECT LOCALTIMESTAMP FROM dual | Return a TIMESTAMP value that represents the current date and time in the session time zone. | |
| [MONTHS\_BETWEEN](https://www.oracletutorial.com/oracle-date-functions/oracle-months_between/) | MONTHS\_BETWEEN( DATE '2017-07-01', DATE '2017-01-01' ) | Return the number of months between two dates. | |
| [NEW\_TIME](https://www.oracletutorial.com/oracle-date-functions/oracle-new_time/) | NEW\_TIME( TO\_DATE( '08-07-2017 01:30:45', 'MM-DD-YYYY HH24:MI:SS' ), 'AST', 'PST' ) | Convert a date in one time zone to another | |
| [NEXT\_DAY](https://www.oracletutorial.com/oracle-date-functions/oracle-next_day/) | NEXT\_DAY( DATE '2000-01-01', 'SUNDAY' ) | Get the first weekday that is later than a specified date. | |
| [ROUND](https://www.oracletutorial.com/oracle-date-functions/oracle-round/) | ROUND(DATE '2017-07-16', 'MM') | Return a date rounded to a specific unit of measure. | |
| [SESSIONTIMEZONE](https://www.oracletutorial.com/oracle-date-functions/oracle-sessiontimezone/) | SELECT SESSIONTIMEZONE FROM dual; | Get the session time zone | |
| [SYSDATE](https://www.oracletutorial.com/oracle-date-functions/oracle-sysdate/) | SYSDATE | Return the current system date and time of the operating system where the Oracle Database resides. | |
| [SYSTIMESTAMP](https://www.oracletutorial.com/oracle-date-functions/oracle-systimestamp/) | SELECT SYSTIMESTAMP FROM dual; | Return the system date and time that includes fractional seconds and time zone. | |
| [TO\_CHAR](https://www.oracletutorial.com/oracle-date-functions/oracle-to_char/) | TO\_CHAR( DATE'2017-01-01', 'DL' ) | Convert a [DATE](https://www.oracletutorial.com/oracle-basics/oracle-date/) or an [INTERVAL](https://www.oracletutorial.com/oracle-basics/oracle-interval/) value to a character string in a specified format. | |
| [TO\_DATE](https://www.oracletutorial.com/oracle-date-functions/oracle-to_date/) | TO\_DATE( '01 Jan 2017', 'DD MON YYYY' ) | Convert a date which is in the character string to a DATE value. | |
| [TRUNC](https://www.oracletutorial.com/oracle-date-functions/oracle-trunc/) | TRUNC(DATE '2017-07-16', 'MM') | Return a date truncated to a specific unit of measure. | |
| [TZ\_OFFSET](https://www.oracletutorial.com/oracle-date-functions/oracle-tz_offset/) | TZ\_OFFSET( 'Europe/London' ) | Get time zone offset of a time zone name from UTC | |

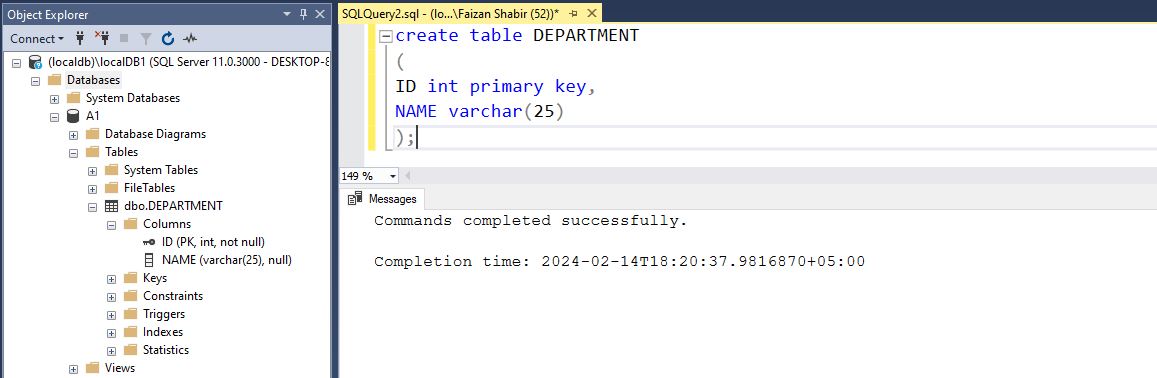
SQL server

* CURRENT\_TIMESTAMP: Returns the current system date and time without the time zone part.
* GETUTCDATE: Returns a date part of a date as an integer number.
* GETDATE: Returns the current system date and time of the operating system on which the SQL Server is running.
* SYSDATETIME: Returns the current system date and time with more fractional seconds precision than the GETDATE() function.
* SYSUTCDATETIME: Returns the current system date and time in UTC time
* SYSDATETIMEOFFSET: Returns the current system date and time with the time zone.
* DATENAME: Returns a date part of a date as a character string
* DATEPART: Returns a date part of a date as an integer number
* DAY: Returns the day of a specified date as an integer
* MONTH: Returns the month of a specified date as an integer
* YEAR: Returns the year of the date as an integer.
* DATEDIFF: Returns a difference in date part between two dates.
* DATEADD: Adds a value to a date part of a date and return the new date value.
* EOMONTH: Returns the last day of the month containing the specified date, with an optional offset.
* WITCHOFFSET: Changes the time zone offset of a DATETIMEOFFSET value and preserves the UTC value.
* TODATETIMEOFFSET: Transforms a DATETIME2 value into a DATETIMEOFFSET value.
* DATEFROMPARTS: Return a DATE value from the year, month, and day.
* DATETIME2FROMPARTS: Returns a DATETIME2 value from the date and time arguments
* DATETIMEOFFSETFROMPARTS: Returns a DATETIMEOFFSET value from the date and time arguments
* TIMEFROMPARTS: Returns a TIME value from the time parts with the precisions
* ISDATE: Check if a value is a valid date, time, or datetime value

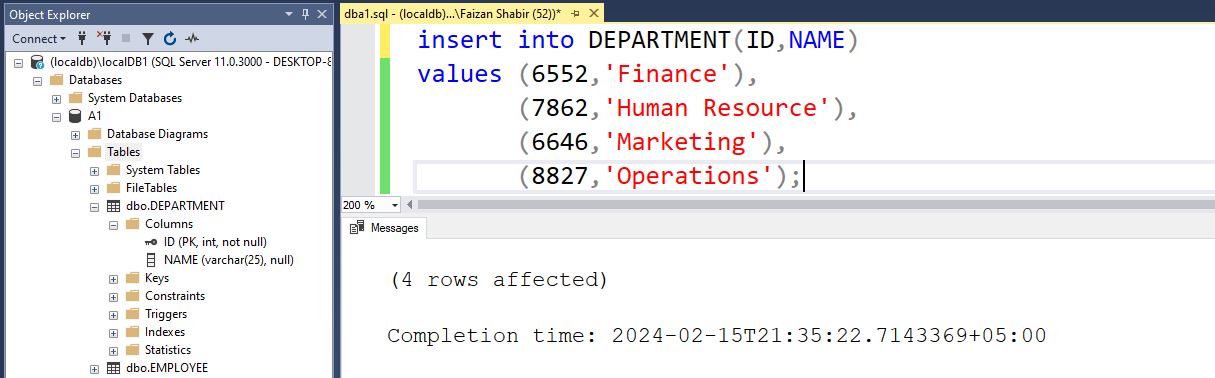
**Part 2**

**Q0: Creating database named A1 **

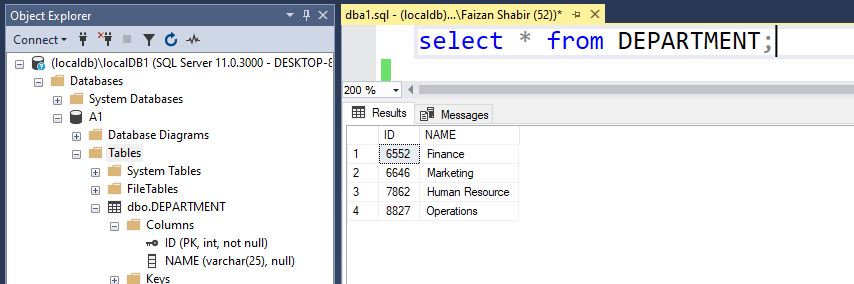
**Q1: Creating table named DEPARTMENT**

****

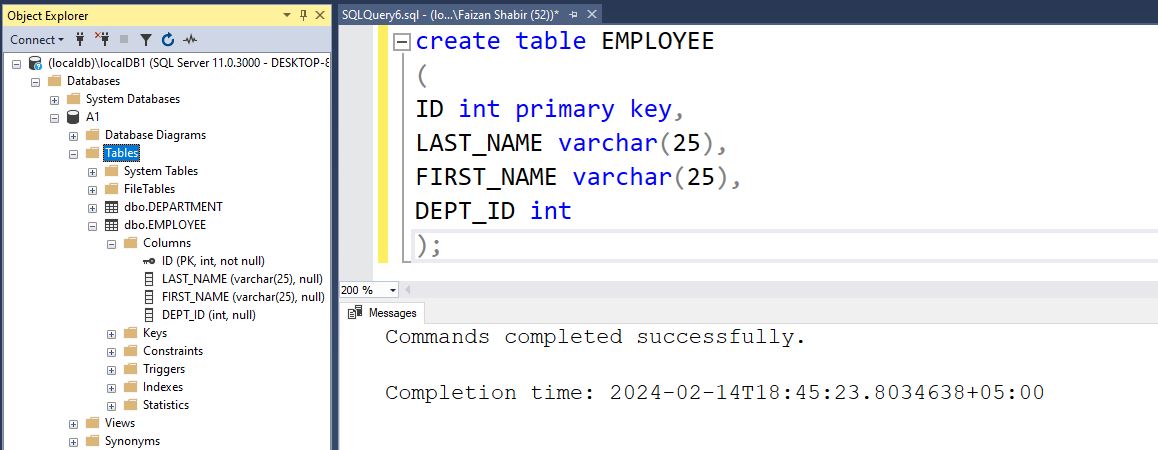
**Q2P1: Populating the DEPARTMENT table with some data.**

****

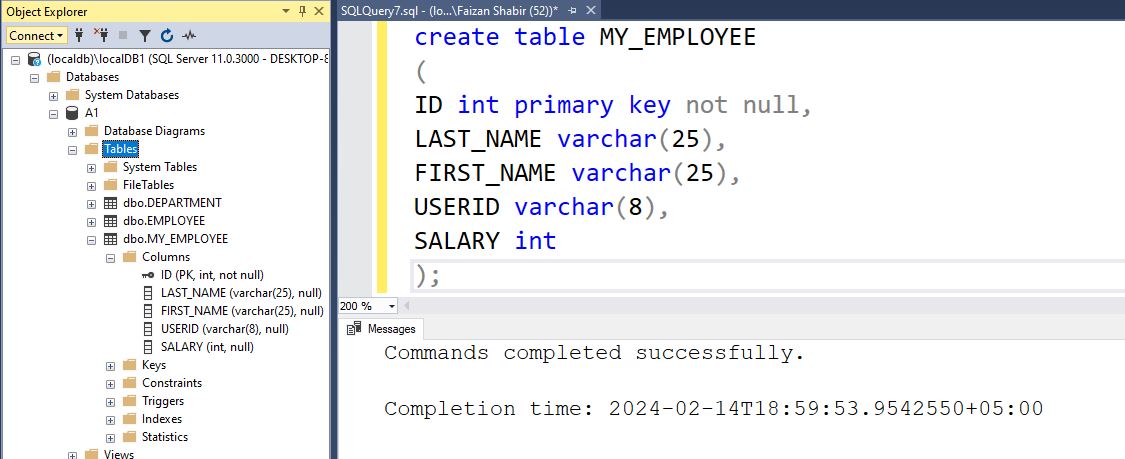
**Q2P2: Outputting the inserted data in the DEPARTMENT table.**

****

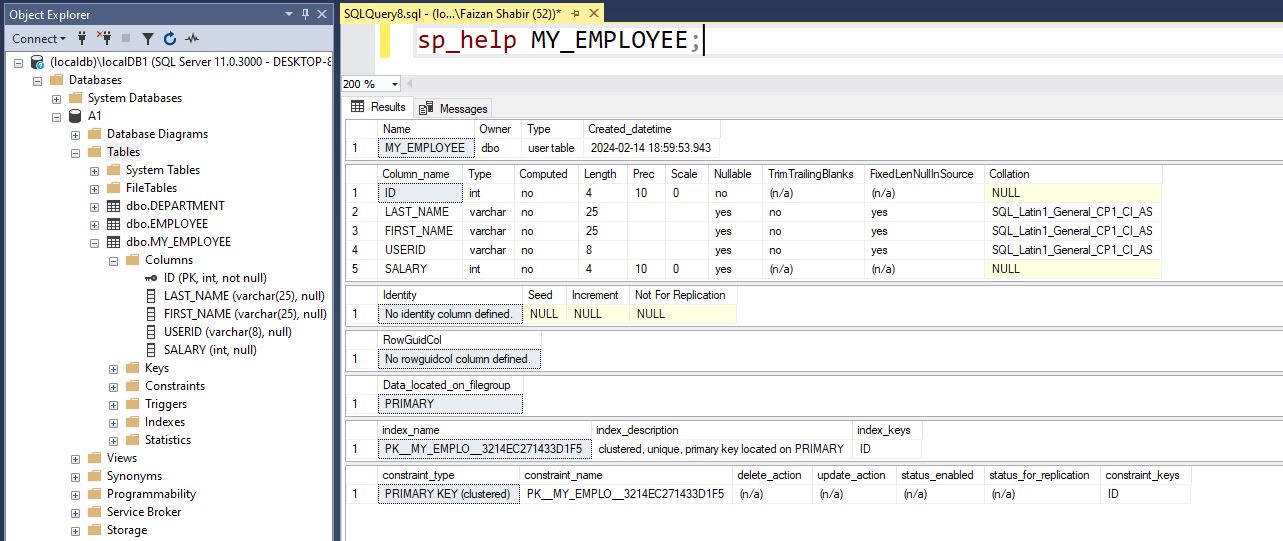
**Q3: Creating the table named EMPLOYEE**

****

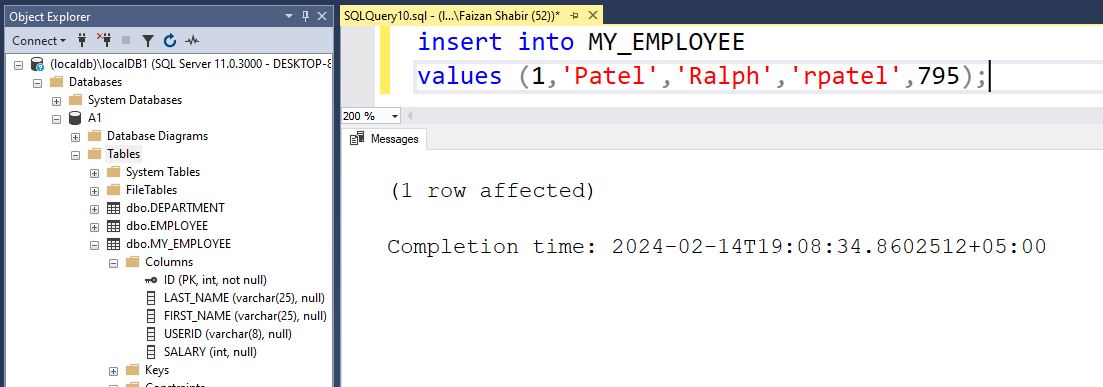
**Q4P1: Creating the table named MY\_EMPLOYEE**

****

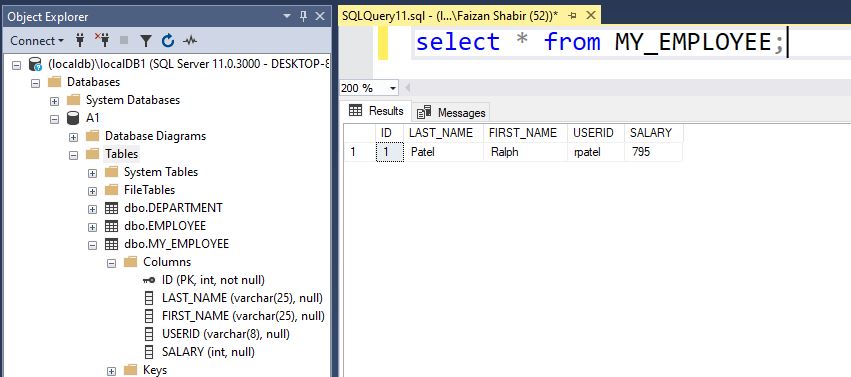
**Q4P2:** **Describing MY\_EMPLOYEE table to identify the column names.**

****

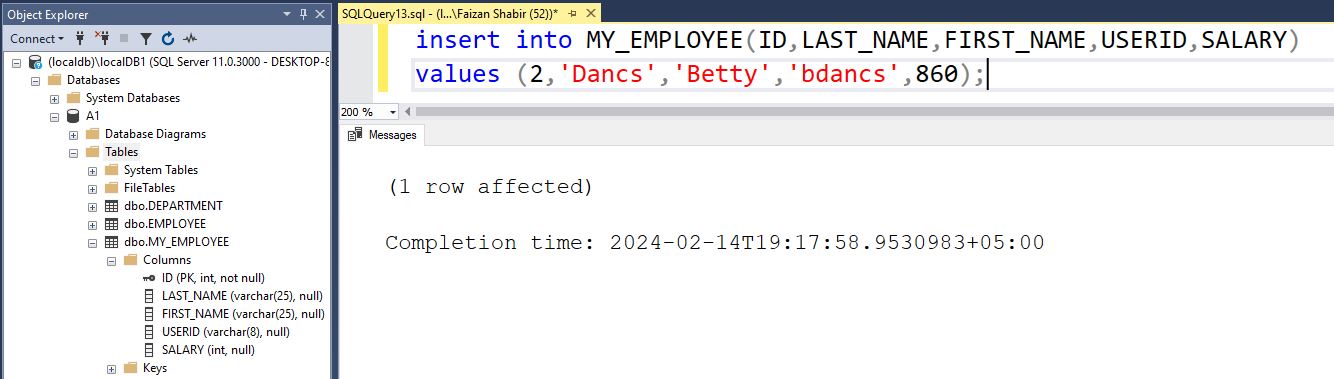
**Q5P1: Populating the MY\_EMPLOYEE table with the first row of sample data.**

****

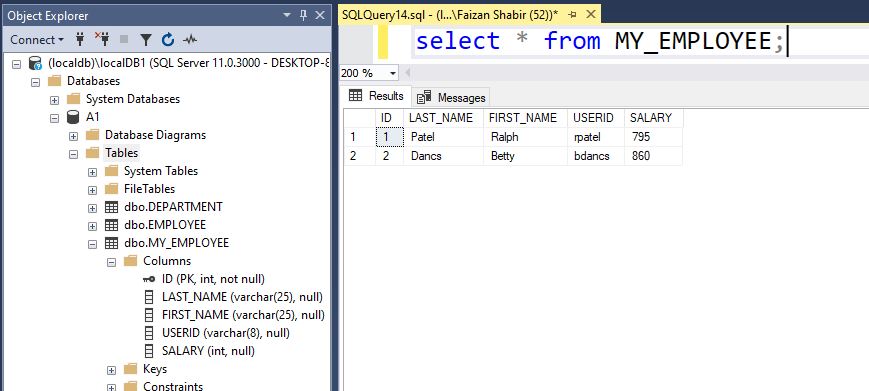
**Q5P2: Outputting the inserted row in the MY\_EMPLOYEE table.**

****

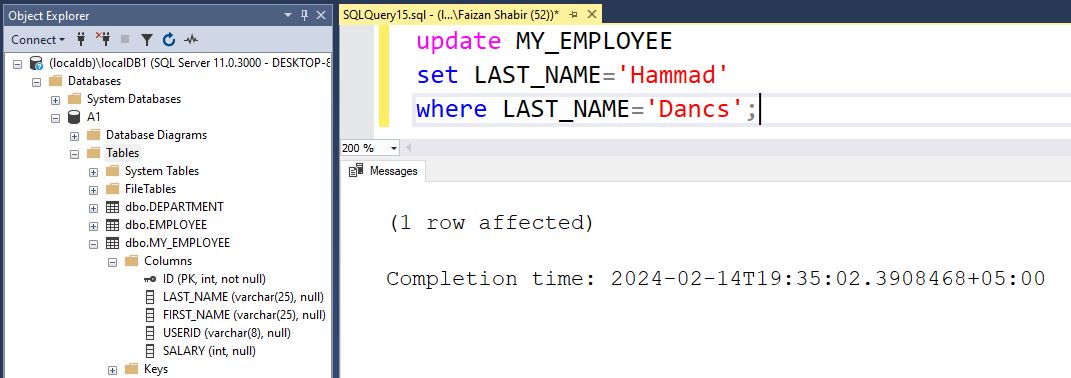
**Q6: Populating the MY\_EMPLOYEE table with the second row of sample data.**

****

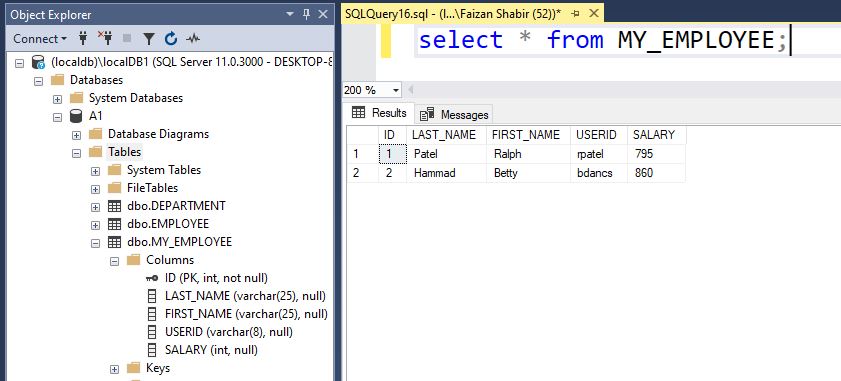
**Q7: Confirming the addition to the table MY\_EMPLOYEE.**

****

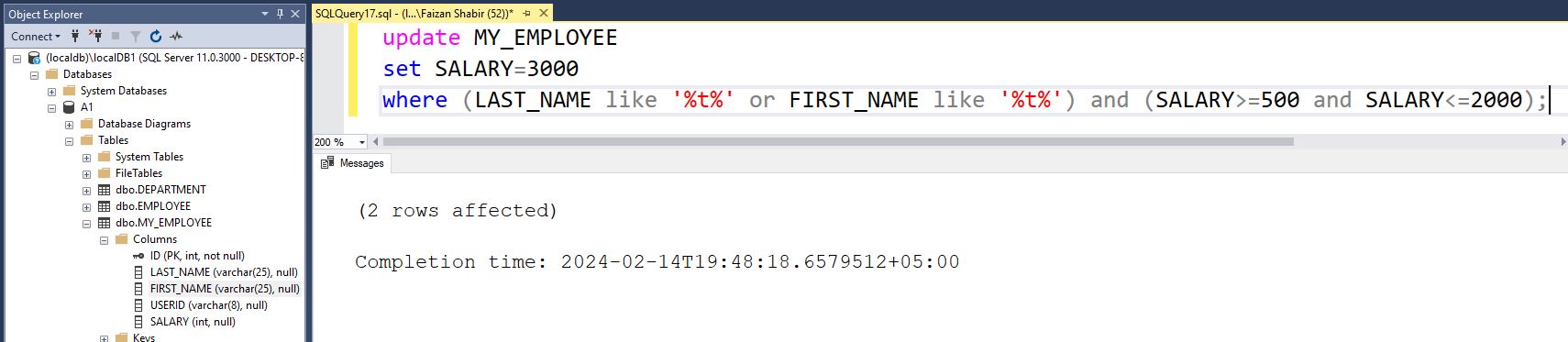
**Q8P1: Changing the last name of employee last\_name ‘Dancs’ to ‘Hammad’**

****

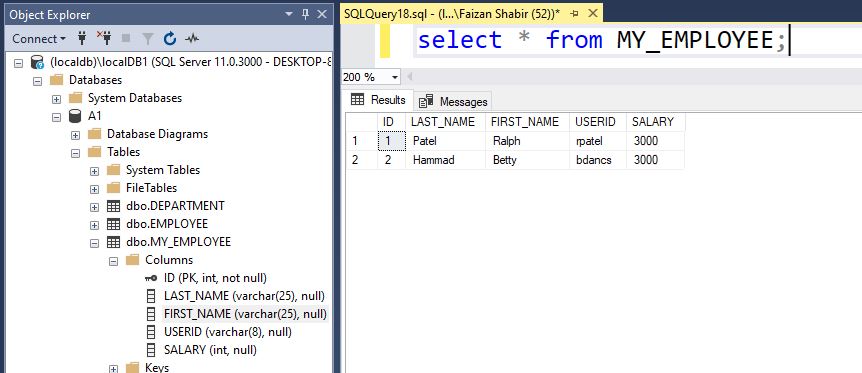
**Q8P2: Confirming the updation done in Q8P1.**

****

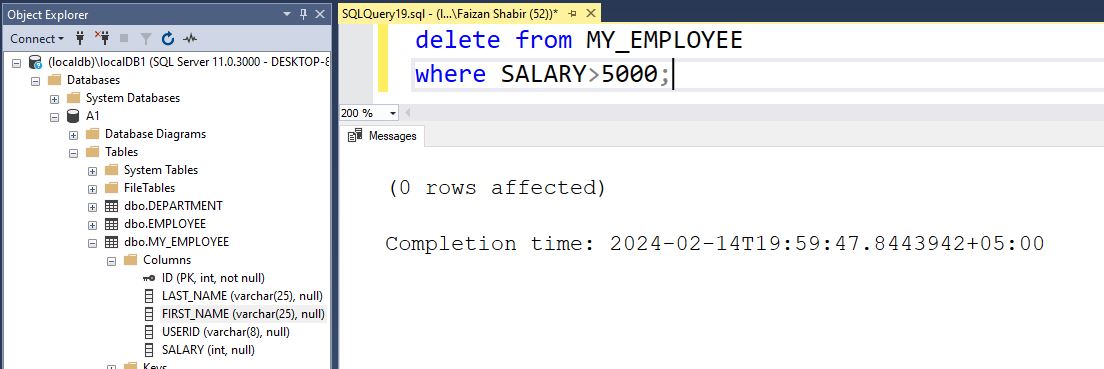
**Q9P1: Changing the salary of all those employees who have a letter‘t’ in their name with salary between 500-2000 to 3000.**

****

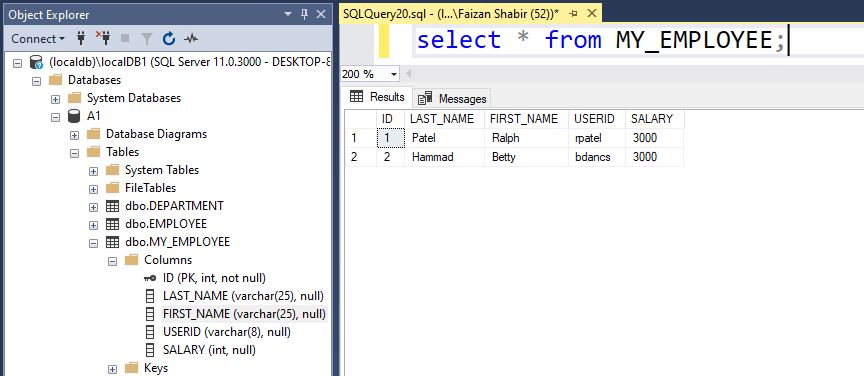
**Q9P2: Confirming the updation done in Q9P1.**

****

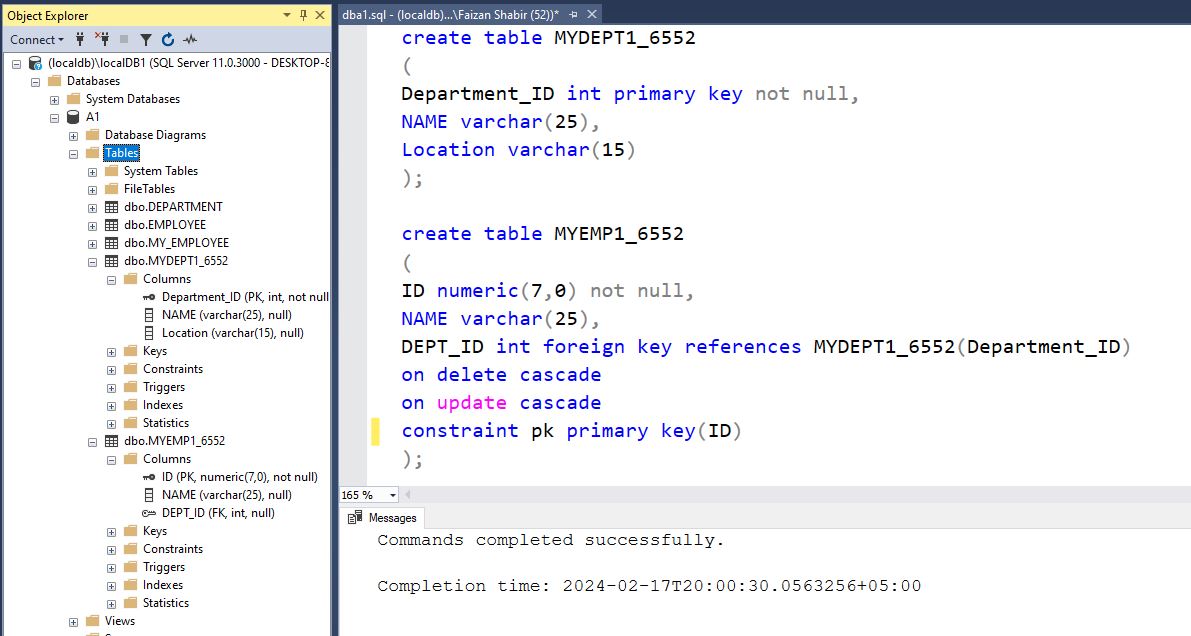
**Q10P1: Deleting an employee who have salary greater than 5000.**

****

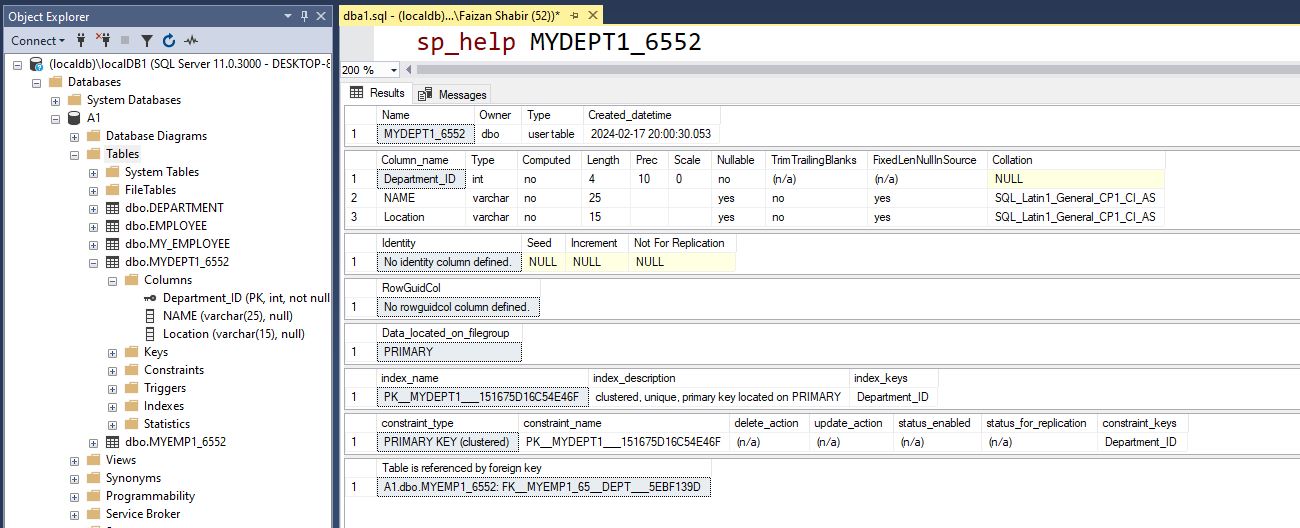
**Q10P2: Confirming the updation done in Q10P1**

****

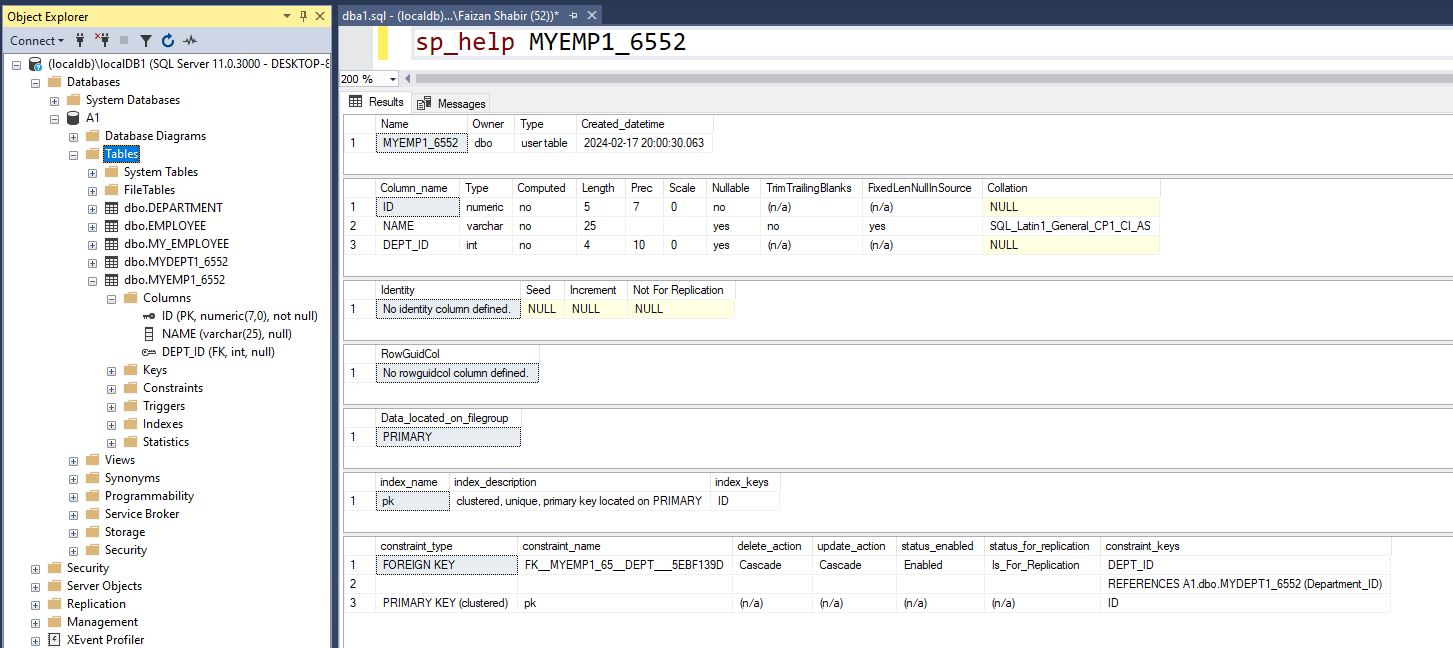
**Q11P1: Creating table MYEMP1\_6552 and MYDEPT1\_6552.**

****

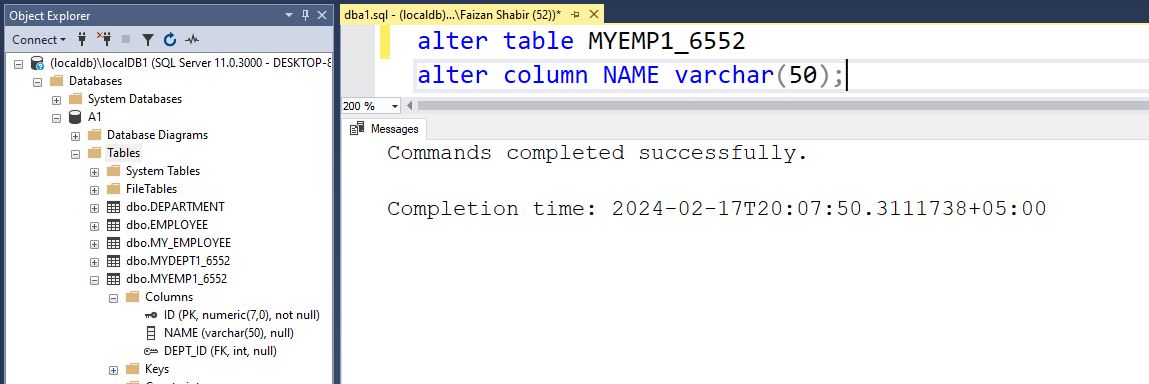
**Q11P2: Confirming that MYDEPT1\_6552 table is created**

****

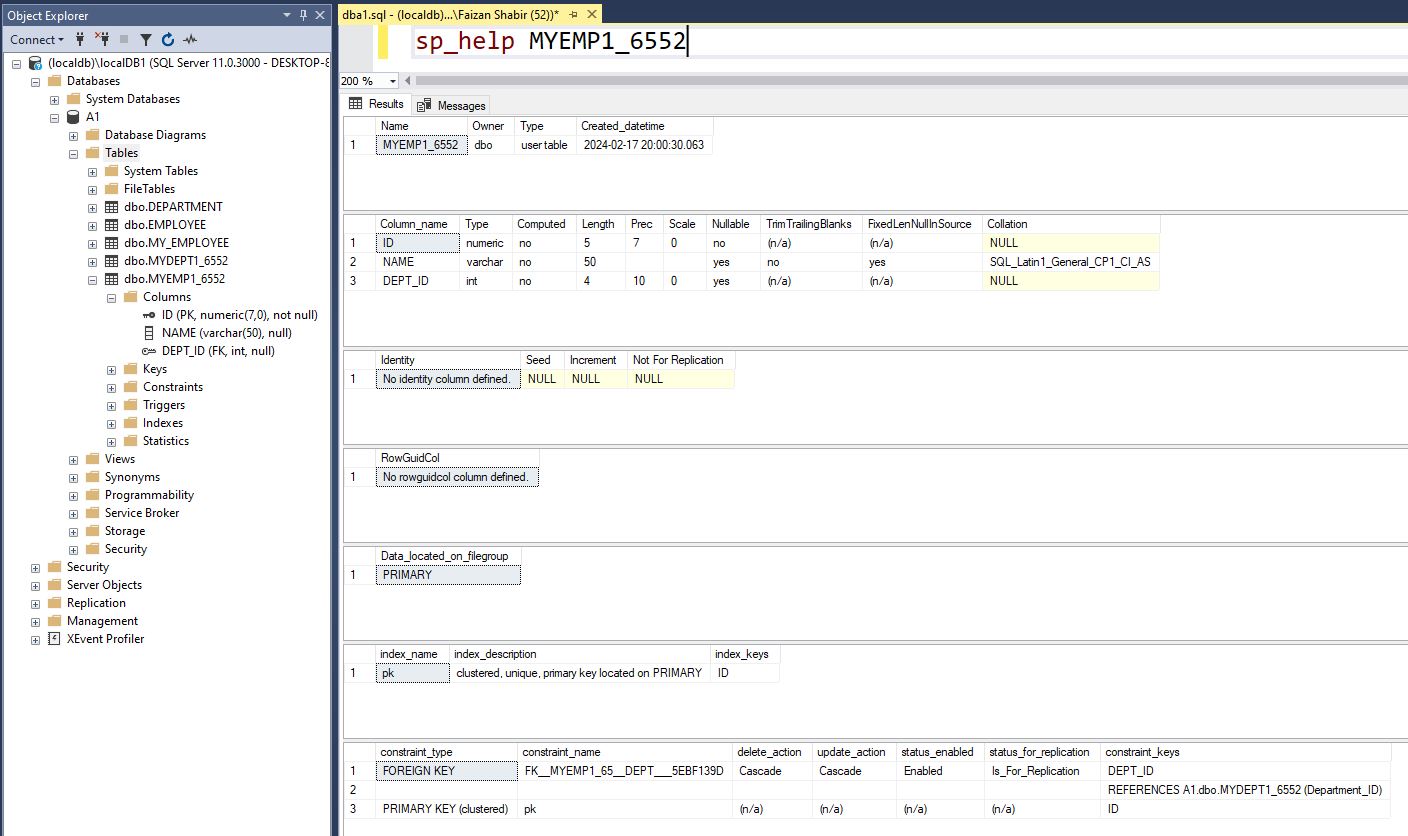
**Q11P3: Confirming that MYEMP1\_6552 table is created**

****

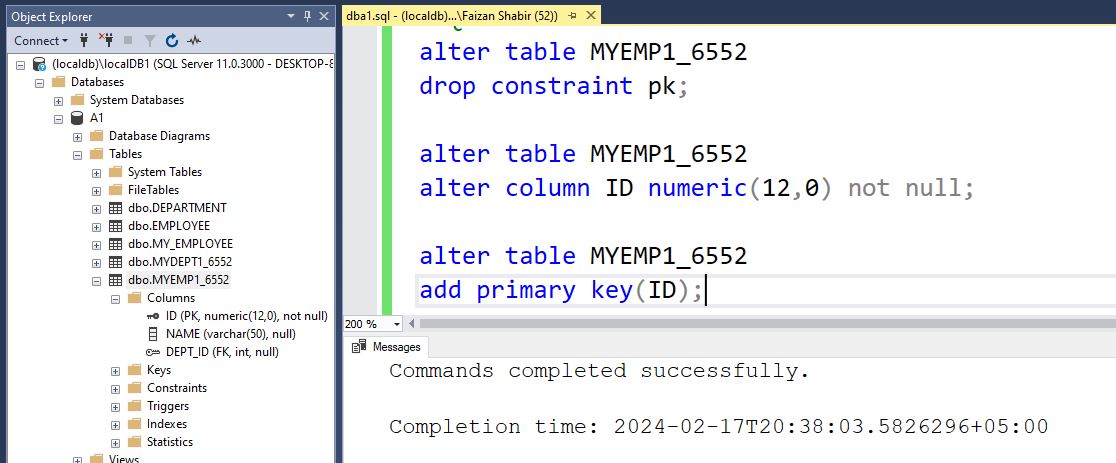
**Q12P1: Modifying the MYEMP1\_6552 table to allow for longer employee names (from 25 characters to 50 characters).**

****

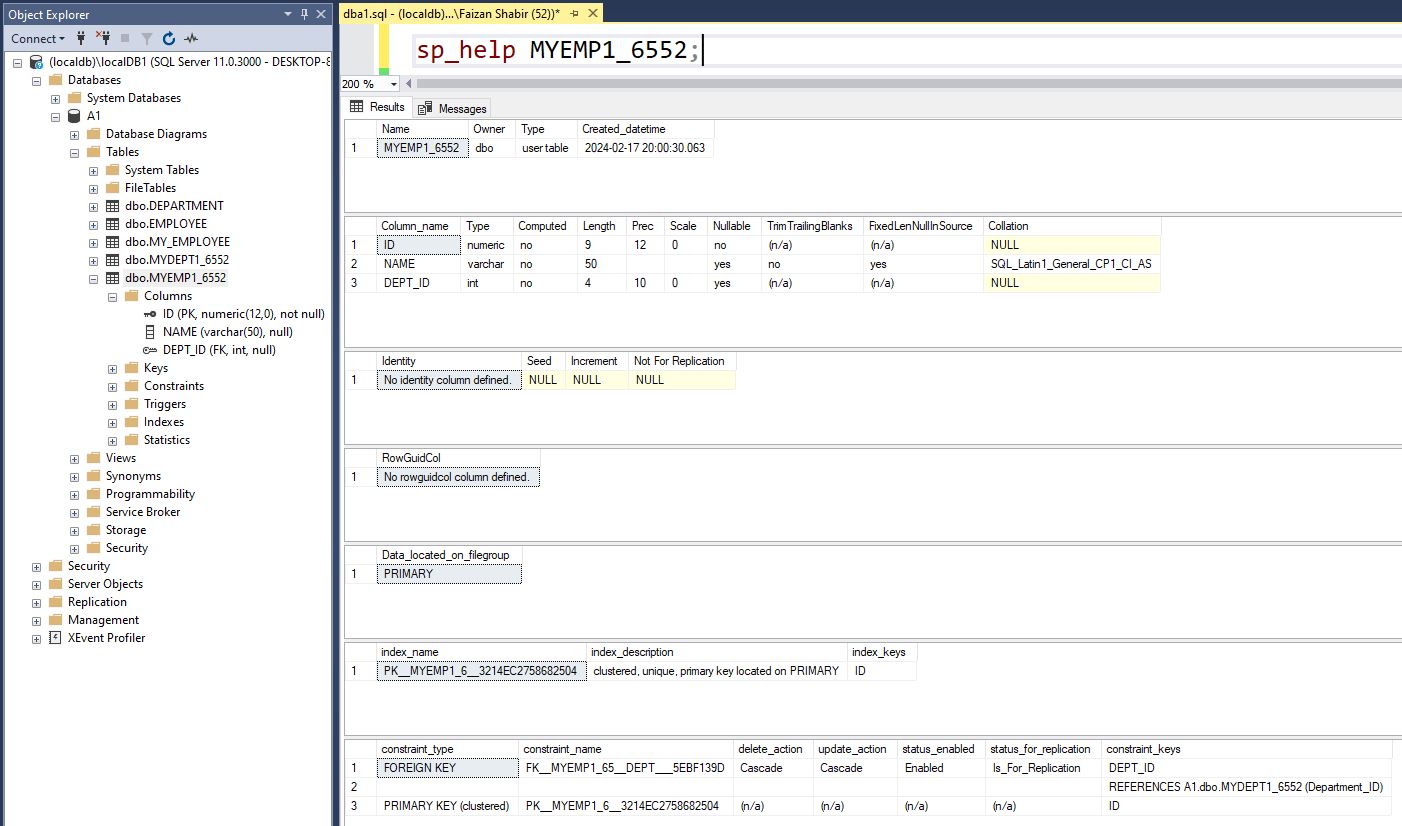
**Q12P2: Confirming the modification done in Q12P1**

****

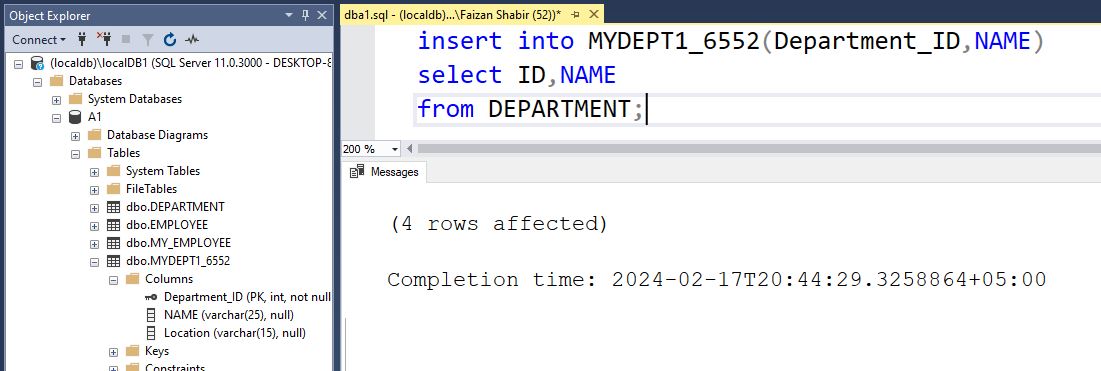
**Q13P1: Modifying the MYEMP1\_6552 table to allow for employee ID to length 12.**

****

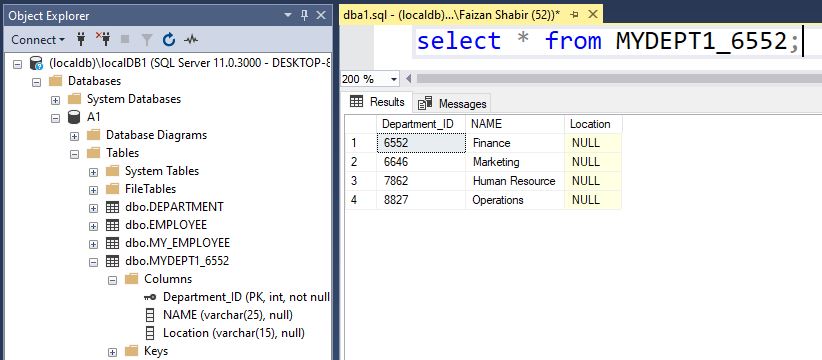
**Q13P2: Confirming the modification done in Q13P1**

****

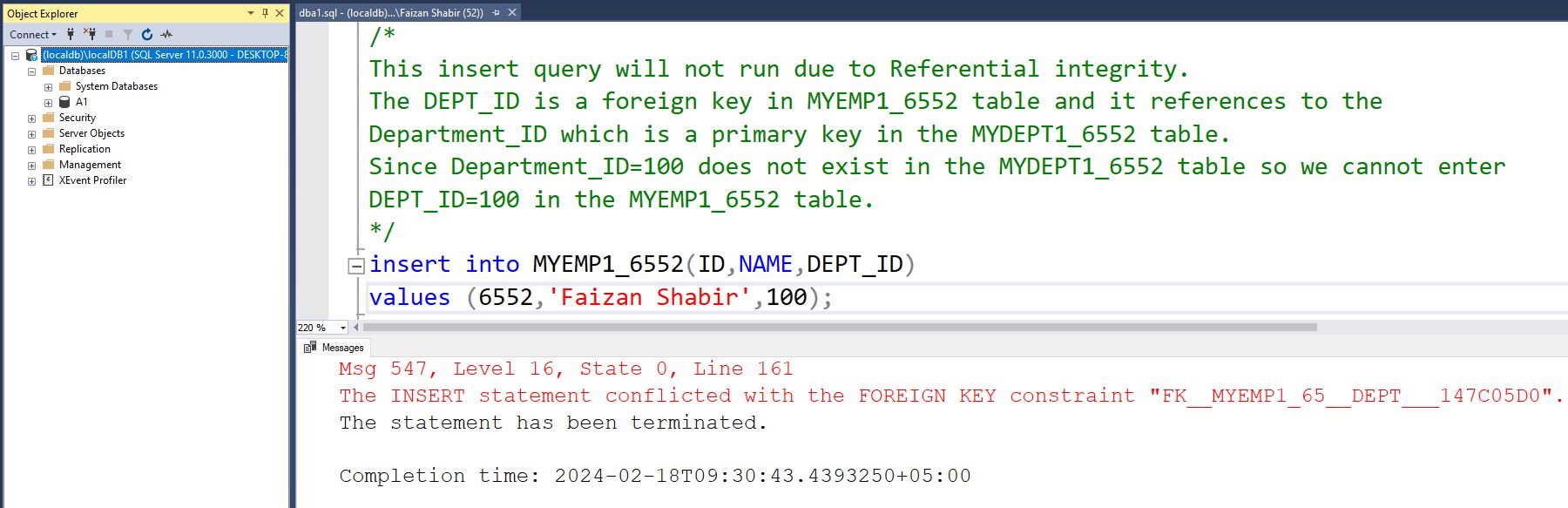
**Q14P1: Populating MYDEPT1\_ 6552 with data from the DEPT table.**

****

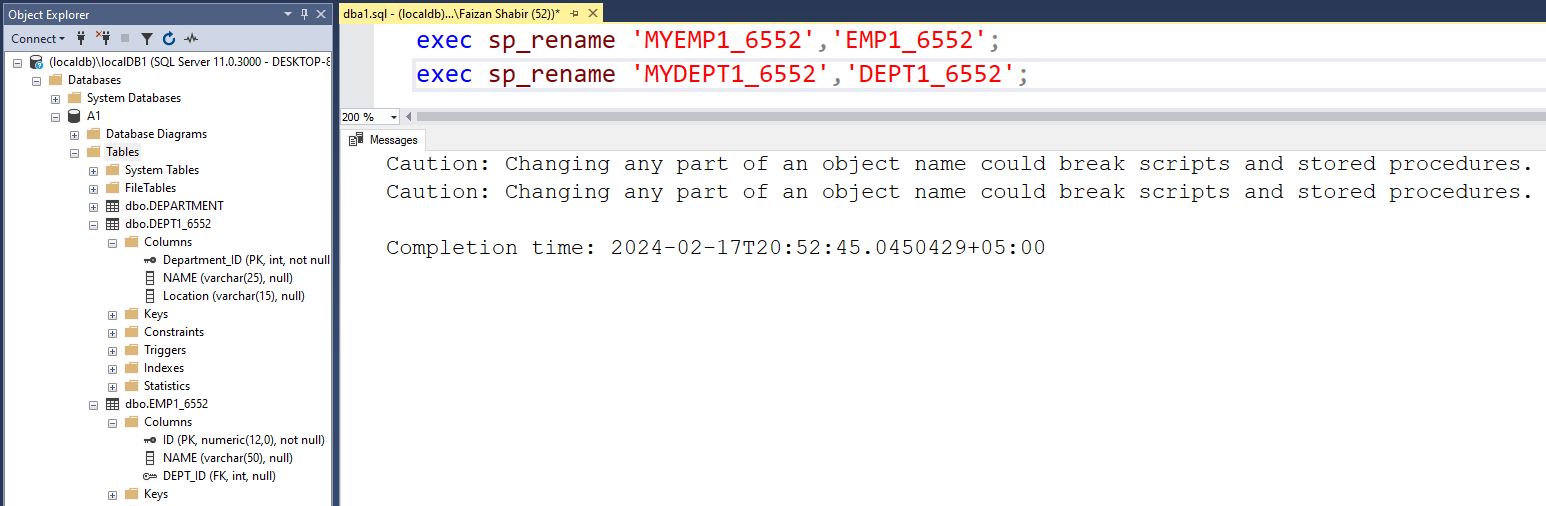
**Q14P2: Confirming the insertion done in Q14P2**

****

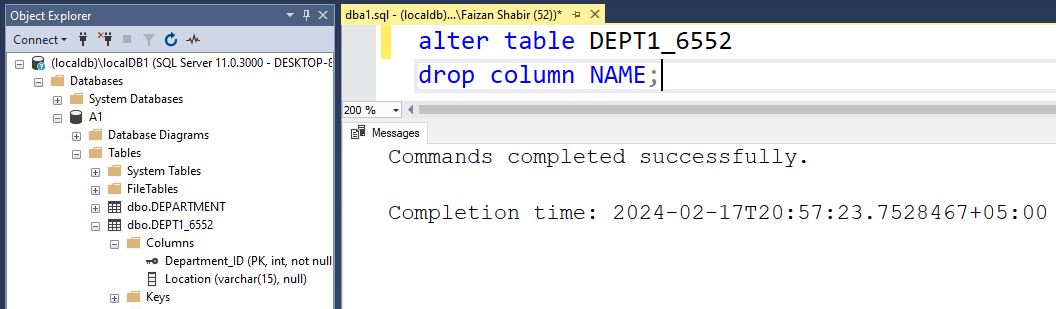
**Q15: Inserting a row in MYEMP1\_ 6552 with id=6552, name=Faizan Shabir and dept\_id = 100 (In both cases 1) inserted 2) not inserted write reason)**

****

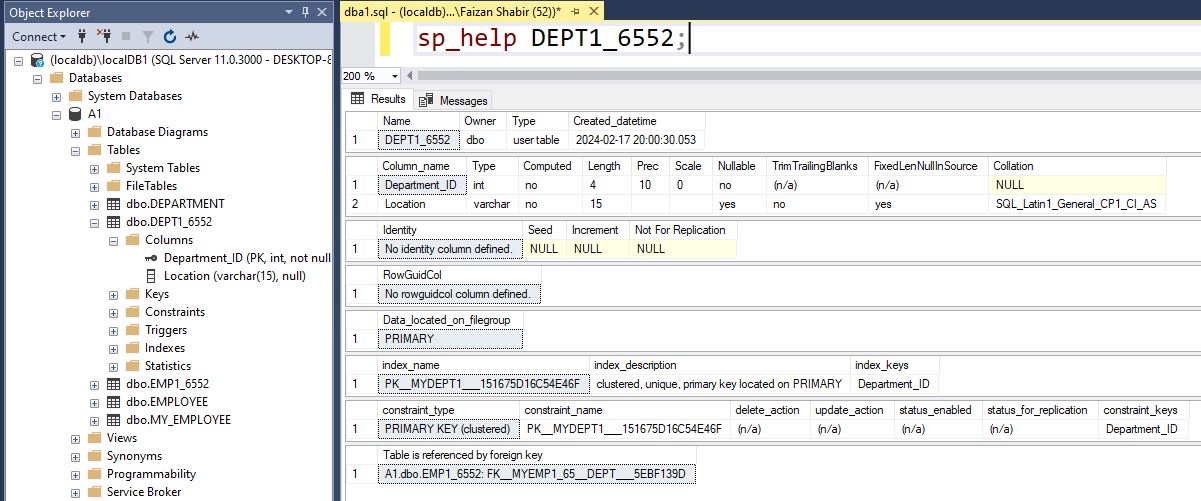
**Q16: Renaming the MYEMP1\_ 6552 table to EMP1\_ 6552 and MYDEPT1\_6552 to DEPT1\_ 6552**

****

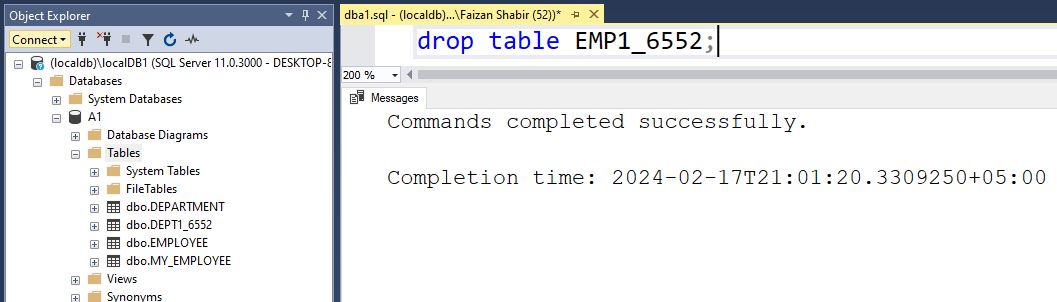
**Q17P1: Dropping the NAME column from the DEPT1\_ 6552 table.**

****

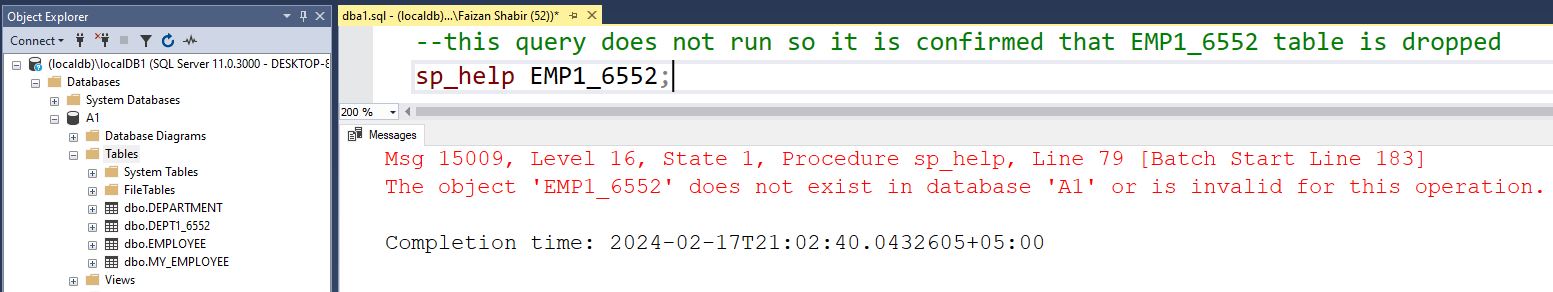
**Q17P2: Confirming the modification done in Q17P1**

****

**Q18P1: Dropping the EMP1\_ 6552 table.**

****

**Q18P2: Confirming the modification done in Q18P1**

****