SQL Server

Assignment

Submitted by : Vijit

Code:

```
use KANINIBATCH2;

CREATE TABLE STUDENT(STUDENT_ID INT PRIMARY KEY, NAME
VARCHAR(20),COURSEID INT,CITY VARCHAR(15));

CREATE TABLE course (course_id INT PRIMARY KEY,course_name VARCHAR(50) NOT
NULL,credits INT);

CREATE TABLE teacher (teacher_id INT PRIMARY KEY,teacher_name VARCHAR(50) NOT
NULL,main_course_id INT);

INSERT INTO STUDENT  VALUES

(101, 'Rahul Sharma', 101, 'Mumbai'),

(102, 'Priya Singh', 103, 'Delhi'),

(103, 'Amit Gupta', 101, 'Bengaluru'),

(104, 'Sneha Kumari', 102, 'Chennai'),

(105, 'Vikas Reddy', 105, 'Hyderabad'),

(106, 'Anjali Devi', 103, 'Mumbai'),

(107, 'Arjun Patel', 106, 'Delhi');


INSERT INTO course  VALUES

(101, 'Database Management', 4),

(102, 'Web Development', 3),

(103, 'Data Science', 5),

(104, 'Cyber Security', 4);

insert into course values(107, 'UI', 4);

delete from course where course_id = 107;


INSERT INTO teacher  VALUES
```

```
(1, 'Dr. Anand Kumar', 101),

(2, 'Prof. Sarita Rao', 103),

(3, 'Mr. Deepak Mehta', 102),

(4, 'Ms. Preeti Verma', 104),

(5, 'Dr. Rajeev Sinha', 999);

ALTER TABLE STUDENT

ADD CONSTRAINT FK_STUDENT_COURSE1 FOREIGN KEY (COURSEID)

REFERENCES course (course_id); --THIS DID NOT WORK AS THE DATA IS
INCONSISTENT. IN ORDER TO ADD A FK CONSTRAINT, DATA HAS TO BE CONSISTENT.


SELECT name, city FROM student;

SELECT name, city

FROM student

WHERE city = 'Mumbai'; --USING THE WHERE CLAUSE


SELECT course_id

from course

where credits>3


SELECT          -- Using inner join. It displays only the common columns in the tables
that are using inner join. The data that is not matched to other tables is left out.In this
case student with id 105 is left.

  s.STUDENT_ID as sid,

  s.name AS StudentName,

  c.course_name AS CourseName

FROM

  student s

INNER JOIN

  course c ON s.courseid = c.course_id;
```

```
SELECT                    --In this case teacher with courseid 999 is not in the result as there is no course in course table with 999 id.
  t.teacher_name AS TeacherName,
  c.course_name AS MainCourseTaught
FROM
  teacher t
INNER JOIN
  course c ON t.main_course_id = c.course_id;


SELECT                    -- Selecting the StudentName, City, and CourseName for all students who are linked to an existing course.
  s.name as STUDENT_NAME,
  s.CITY,
  c.course_name
FROM
  STUDENT s
INNER JOIN
  course c ON s.courseid = c.course_id


SELECT --Usiing left join to get the vakues that are not linked to other values as well in the result.
  s.name AS StudentName,
  s.city,
  c.course_name AS CourseName,
  c.credits
FROM
  student s
LEFT JOIN
```

```
    course c ON s.courseid = c.course_id;


SELECT
    t.teacher_name AS TeacherName,
    c.course_name AS MainCourseTaught
FROM
    teacher t
LEFT JOIN
    course c ON t.main_course_id = c.course_id;


SELECT
    t.teacher_name AS TeacherName,
    c.course_name AS MainCourseTaught
FROM
    course c
LEFT JOIN
    teacher t ON c.course_id = t.main_course_id;


SELECT --Usiing left join to get the vakues that are not linked to other values as well in the result.
    s.name AS StudentName,
    s.city,
    c.course_name AS CourseName,
    c.credits
FROM
    student s
LEFT JOIN
    course c ON s.courseid = c.course_id
```

```
where
   c.course_name is null;

SELECT --Right join to display student data
   c.course_name,
   s.name AS StudentName
FROM
   student s
RIGHT JOIN
   course c ON s.courseid = c.course_id;

SELECT
   c.course_name AS CourseName,
   t.teacher_name AS TeacherName
FROM
   course c
RIGHT JOIN
   teacher t ON c.course_id = t.main_course_id;

SELECT
   s.NAME AS Sname,
   c.course_name  AS Cname
from
   course c
right join
   STUDENT s on s.courseid = c.course_id;

SELECT
```

```
  t.teacher_name as TEACHER,
  c.course_name
FROM
  course c
RIGHT JOIN
  teacher t ON c.course_id = t.main_course_id
WHERE
  c.course_id is null;


SELECT     --Full join. It returns all the data from both the tables. It is like the combination of inner join, left and right join.
   s.name AS StudentName,
   s.city,
   s.courseid AS StudentAssignedCourseID,
   c.course_name AS CourseName,
   c.credits AS CourseCredits,
   c.course_id AS CourseIDInTable
FROM
   student s
FULL OUTER JOIN
   course c ON s.courseid = c.course_id;
SELECT
   s.name AS StudentName,
   s.city,
   s.courseid AS StudentAssignedCourseID,
   c.course_name AS CourseName,
   c.credits AS CourseCredits,
   c.course_id AS CourseIDInTable
```

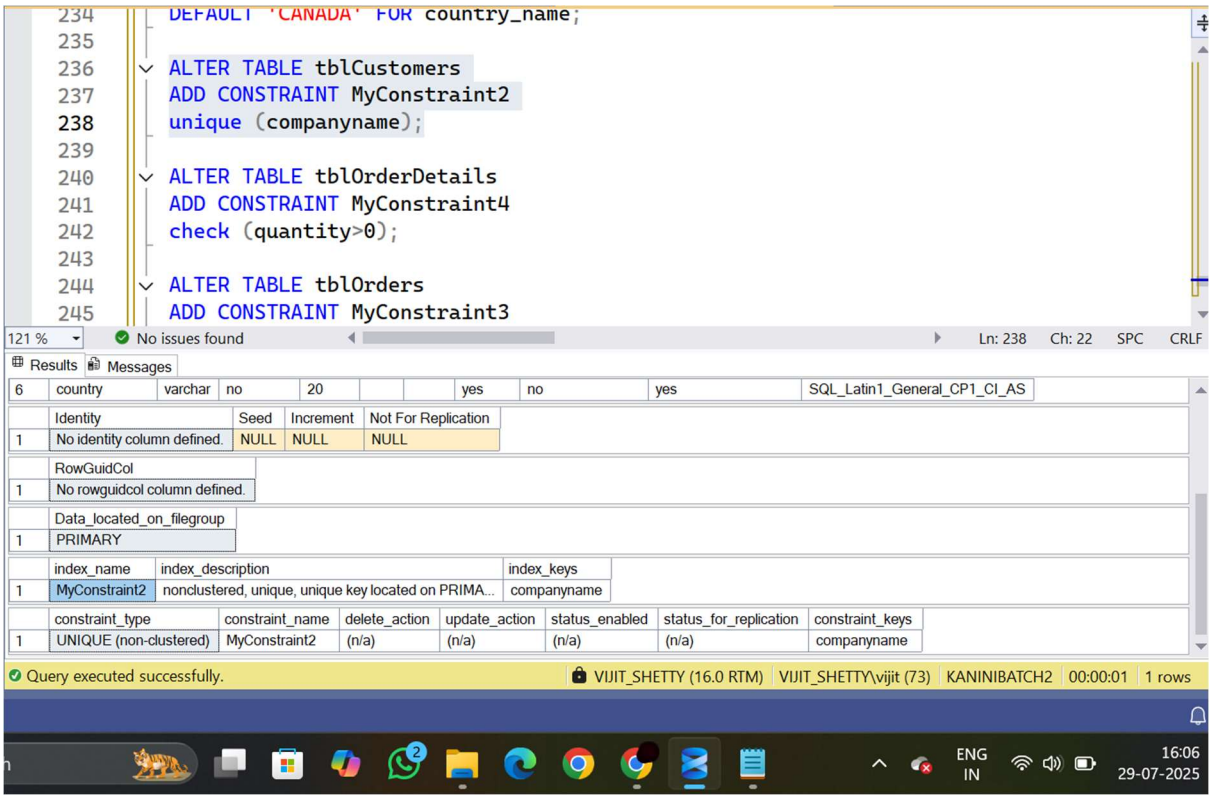
```
FROM
  student s
FULL OUTER JOIN
  course c ON s.courseid = c.course_id
WHERE
  s.COURSEID IS NULL OR c.course_id is null;

SELECT      -- Subqueries practice with IN keyword
  s.name,
  s.city,
  s.courseid
FROM
  student s
WHERE
  s.courseid IN (SELECT course_id FROM course WHERE credits > 4);

SELECT
  c.course_name,
  c.credits
FROM
  course c
WHERE
  c.course_id IN (SELECT main_course_id FROM teacher WHERE teacher_id > 3);

SELECT
  s.name,
  s.city,
  s.courseid
```

```
FROM
  student s
WHERE
  s.COURSEID not in (select course_id from course where credits = 3);

SELECT  --Using exists with subquery.
  s.name,
  s.city
FROM
  student s
WHERE
  EXISTS (SELECT 1 FROM course c WHERE c.course_id = s.courseid);

SELECT
  c.course_name,
  c.credits
FROM
  course c
WHERE
  EXISTS (SELECT 1 FROM student s WHERE s.courseid = c.course_id);

SELECT
  t.teacher_name AS NAME
FROM
  teacher t
WHERE
  EXISTS (SELECT 1
```

```
        FROM course c
        WHERE c.course_id = t.main_course_id
         AND c.credits > 4);


SELECT
   c.course_id AS cid,
   c.course_name AS cname
FROM
   course c
WHERE
   NOT EXISTS (SELECT 1
          FROM teacher t
          WHERE t.main_course_id = c.course_id
          );
--Assignment
create table tblShippers(shopperid int,country_name varchar(20));
create table tblorderDetails(orderid int, productid int,unitprice int, quantity int, discount int);
create table tblCustomers(customerid int,companyname varchar(20),contactname varchar(20),contacttitle varchar(15),address varchar(20),country varchar(20));
create table tblOrders(orderid int, customer_id int,employeeid int,orderdate date,reqdate date,shippeddate date);
ALTER TABLE tblShippers
ADD CONSTRAINT MyConstraint
DEFAULT 'CANADA' FOR country_name;


ALTER TABLE tblCustomers
ADD CONSTRAINT MyConstraint2
unique (companyname);
```

```
ALTER TABLE tblOrderDetails

ADD CONSTRAINT MyConstraint4

check (quantity>0);


ALTER TABLE tblOrders

ADD CONSTRAINT MyConstraint3

check (shippeddate > orderdate);


exec sp_help tblShippers;
```

Snapshots:



```
237     ADD CONSTRAINT MyConstraint2
238     unique (companyname);
239
240     ALTER TABLE tblOrderDetails
241     ADD CONSTRAINT MyConstraint4
242     check (quantity>0);
243
244     ALTER TABLE tblOrders
245     ADD CONSTRAINT MyConstraint3
246     check (shippeddate > orderdate);
247
248     exec sp_help tblOrderDetails;
```

121 % | No issues found | Ln: 243 | Ch: 1 | SPC | CRLF

Results | Messages

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 4 | quantity | int | no | 4 | 10 | 0 | yes | (n/a) | (n/a) | NULL |
| 5 | discount | int | no | 4 | 10 | 0 | yes | (n/a) | (n/a) | NULL |

| | Identity | Seed | Increment | Not For Replication |
|---|---|---|---|---|
| 1 | No identity column defined. | NULL | NULL | NULL |

| | RowGuidCol |
|---|---|
| 1 | No rowguidcol column defined. |

| | Data_located_on_filegroup |
|---|---|
| 1 | PRIMARY |

| | constraint_type | constraint_name | delete_action | update_action | status_enabled | status_for_replication | constraint_keys |
|---|---|---|---|---|---|---|---|
| 1 | CHECK on column quantity | MyConstraint1 | (n/a) | (n/a) | Enabled | Is_For_Replication | ([quantity]>(0)) |
| 2 | CHECK on column quantity | MyConstraint4 | (n/a) | (n/a) | Enabled | Is_For_Replication | ([quantity]>(0)) |

Query executed successfully. | VIJIT_SHETTY (16.0 RTM) | VIJIT_SHETTY\vijit (73) | KANINIBATCH2 | 00:00:01 | 11 rows

ENG IN 16:08 29-07-2025

```
243
244     ALTER TABLE tblOrders
245     ADD CONSTRAINT MyConstraint3
246     check (shippeddate > orderdate);
247
248     exec sp_help tblOrders;
```

121 % | No issues found | Ln: 244 | Ch: 1 | SPC | CRLF

Results | Messages

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 4 | orderdate | date | no | 3 | 10 | 0 | yes | (n/a) | (n/a) | NULL |
| 5 | reqdate | date | no | 3 | 10 | 0 | yes | (n/a) | (n/a) | NULL |
| 6 | shippeddate | date | no | 3 | 10 | 0 | yes | (n/a) | (n/a) | NULL |

| | Identity | Seed | Increment | Not For Replication |
|---|---|---|---|---|
| 1 | No identity column defined. | NULL | NULL | NULL |

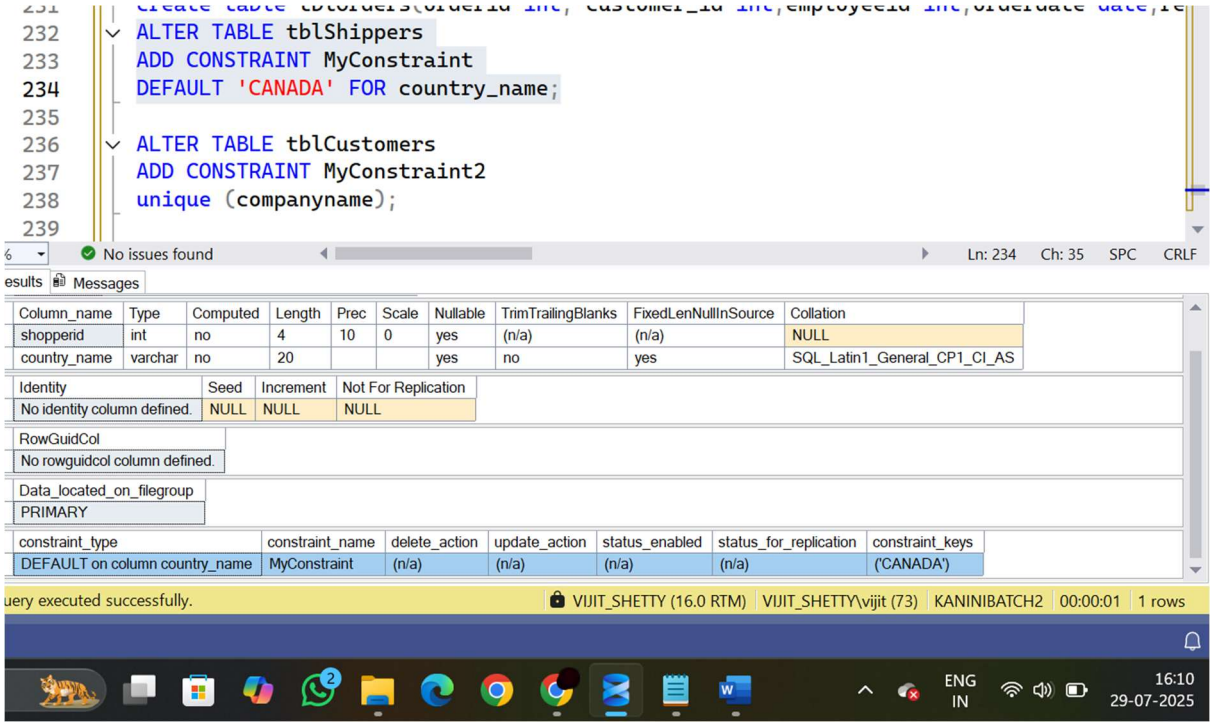
| | RowGuidCol |
|---|---|
| 1 | No rowguidcol column defined. |

| | Data_located_on_filegroup |
|---|---|
| 1 | PRIMARY |

| | constraint_type | constraint_name | delete_action | update_action | status_enabled | status_for_replication | constraint_keys |
|---|---|---|---|---|---|---|---|
| 1 | CHECK Table Level | MyConstraint3 | (n/a) | (n/a) | Enabled | Is_For_Replication | ([shippeddate]>[orderdate]) |

Query executed successfully. | VIJIT_SHETTY (16.0 RTM) | VIJIT_SHETTY\vijit (73) | KANINIBATCH2 | 00:00:01 | 1 rows

ENG IN 16:09 29-07-2025


create table tblOrders(OrderId int, Customer_Id int,EmployeeId int,OrderDate date,Fr
ALTER TABLE tblShippers
ADD CONSTRAINT MyConstraint
DEFAULT 'CANADA' FOR country_name;
ALTER TABLE tblCustomers
ADD CONSTRAINT MyConstraint2
unique (companyname);
No issues found
Ln: 234 Ch: 35 SPC CRLF
esults Messages
Column_name Type Computed Length Prec Scale Nullable TrimTrailingBlanks FixedLenNullInSource Collation
shopperid int no 4 10 0 yes (n/a) (n/a) NULL
country_name varchar no 20 yes no yes SQL_Latin1_General_CP1_CI_AS
Identity Seed Increment Not For Replication
No identity column defined. NULL NULL NULL
RowGuidCol
No rowguidcol column defined.
Data_located_on_filegroup
PRIMARY
constraint_type constraint_name delete_action update_action status_enabled status_for_replication constraint_keys
DEFAULT on column country_name MyConstraint (n/a) (n/a) (n/a) (n/a) ('CANADA')
uery executed successfully.
VIJIT_SHETTY (16.0 RTM) VIJIT_SHETTY\vijit (73) KANINIBATCH2 00:00:01 1 rows
ENG
IN
16:10
29-07-2025