# Department of Computer Science

**CS220: Database Systems**

**Class: BESE-13AB**

# Lab04: DDL and Constraints

# Date: October 02, 2023

# Time: 10:00-1:00 & 02:00-05:00

# Instructor: Dr. Bilal Ali & Dr. Shah Khalid

# Lab Engineer: Sundas Dawood

**Name:** Aimen Munawar

**CMS ID:** 415867

**Class:** BESE-13-A

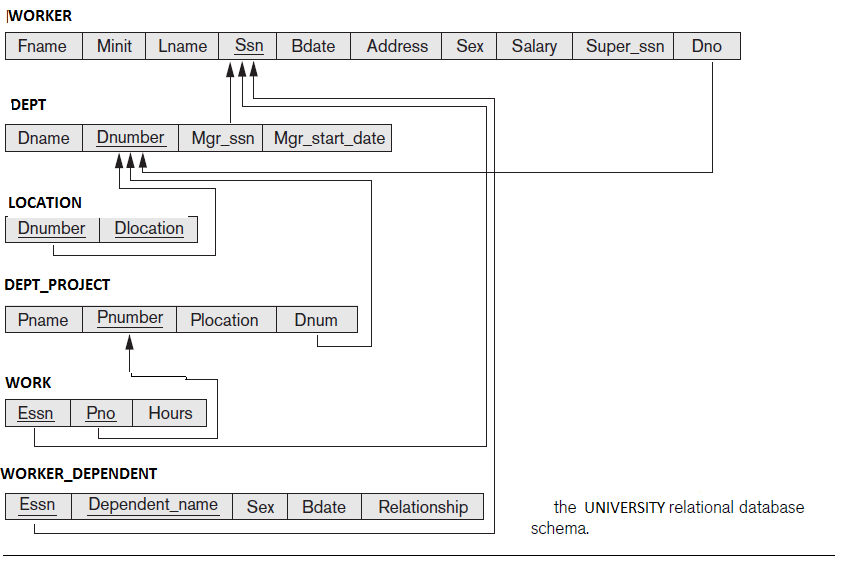
# CLO-1: Create a database schema that incorporates keys and integrity constraints.

# 

**Lab 04: DDL and Constraints**

**Lab Tasks**

**Schema to be implemented:**

****

The COMPANY schema is given in the figure above. Using this schema perform the following tasks:

1. Create COMPANY database and the corresponding tables/relations in the COMPANY schema using the DDL commands. Ensure only the following constraints are specified during the table creation time for Q.1:
   1. Primary Key
   2. Not Null
   3. Unique

**PK values are given in the figure. Bdate attribute should not be null while Dname should always be unique. Assign valid names to all the constraints.**

1. **CODE: Create Database:**

CREATE database COMPANY;

1. **CODE: Creating Tables**

CREATE TABLE worker(

fname varchar(15),

minit varchar(1),

lname varchar(15),

ssn char(9) ,

bdate date not null,

address varchar(30),

sex char,

salary INT,

superssn char(9),

dno INT,

constraint PK\_worker primary key (ssn)

);

CREATE TABLE dept(

dname varchar(15) UNIQUE ,

dnumber INT ,

mgrssn char(9) not null,

mgrstartdate date,

constraint PK\_dept primary key (dnumber)

);

CREATE TABLE locations (

dnumber INT ,

dlocation varchar(15) ,

constraint PK\_locations primary key (dnumber,dlocation)

);

CREATE TABLE dept\_project (

pname varchar(15) not null ,

pnumber INT ,

plocation varchar(15),

dnum INT not null,

constraint PK\_dept\_project primary key (pnumber)

);

CREATE TABLE works (

essn char(9) ,

pno INT ,

hours INT,

constraint PK\_works primary key (essn,pno)

);

CREATE TABLE worker\_dependent(

essn char(9) ,

dependent\_name varchar(15) ,

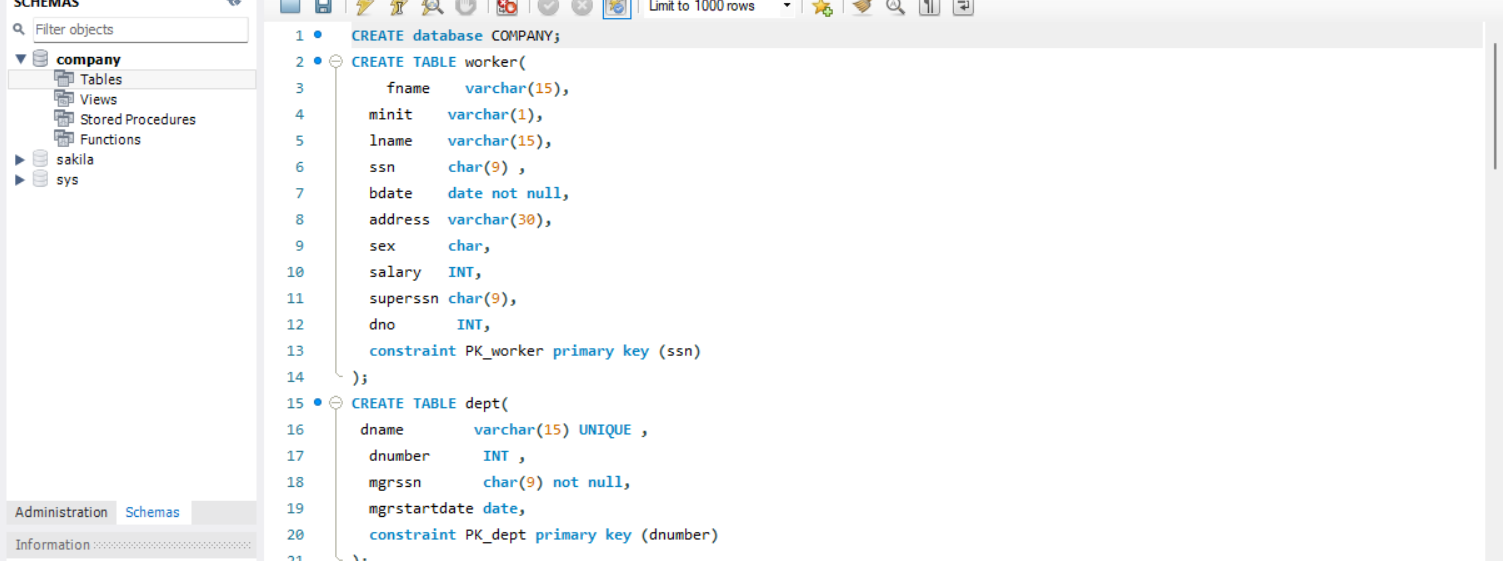
sex char,

bdate date not null,

relationship varchar(20),

constraint PK\_worker\_dependent primary key (dependent\_name,essn)

);



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. After creating the table structures and applying constraints to it, verify if the constraints applied are working as desired (hint: apply constraint violations while inserting updating deleting data).
   1. Insert at least 3 rows of valid data into the tables. Follow the violations which occur while you insert data.

**Inserting 3 rows of Valid data:**

**CODE : Insert rows into Worker table**

INSERT INTO worker

VALUES ('Aimen','1','Munawar',34789,'2022-04-22','50rd Street New york','M','3000','1234',576357);

INSERT INTO worker

VALUES ('Haniya','M','sultan',134578,'2022-03-22','54rd Street Pakistan','M','700','1234',576357);

INSERT INTO worker

VALUES ('Maheen','j','Ali',3789,'2022-06-22','56rd Street UAE','M','3000','124',576357);

**CODE :Insert rows into Department table**

INSERT INTO dept

Values ('English','73579','582467','2023-09-22');

INSERT INTO dept

VALUES ('Mathematics','12345','123456789','2023-10-01');

INSERT INTO dept

VALUES ('Computer','11111','123456789','2023-10-15');

**CODE :Insert rows into Locations table**

INSERT INTO locations

Values('123456','New York');

INSERT INTO locations (dnumber, dlocation)

VALUES ('257687', 'Pakistan');

INSERT INTO locations (dnumber, dlocation)

VALUES ('2547754', 'India');

**CODE :Insert rows into Dept\_project table**

INSERT INTO dept\_project (pname, pnumber, plocation, dnum)

VALUES ('Project D', 101, 'New York', 1);

INSERT INTO dept\_project (pname, pnumber, plocation, dnum)

VALUES ('Project E', 105, 'San Francisco', 5);

INSERT INTO dept\_project (pname, pnumber, plocation, dnum)

VALUES ('Project F', 108, 'Chicago', 3);

**CODE: Insert rows into Worker\_Dependent table**

INSERT INTO worker\_dependent

VALUES ('123456789', 'Child1', 'F', '2010-01-15', 'Child');

INSERT INTO worker\_dependent

VALUES ('333445555', 'Parent', 'M', '2023-12-01', 'Parent');

INSERT INTO worker\_dependent

VALUES ('333445555', 'Spouse1', 'M', '1988-05-20', 'Spouse');

**CODE :Insert rows into Works table**

INSERT INTO works (essn, pno, hours)

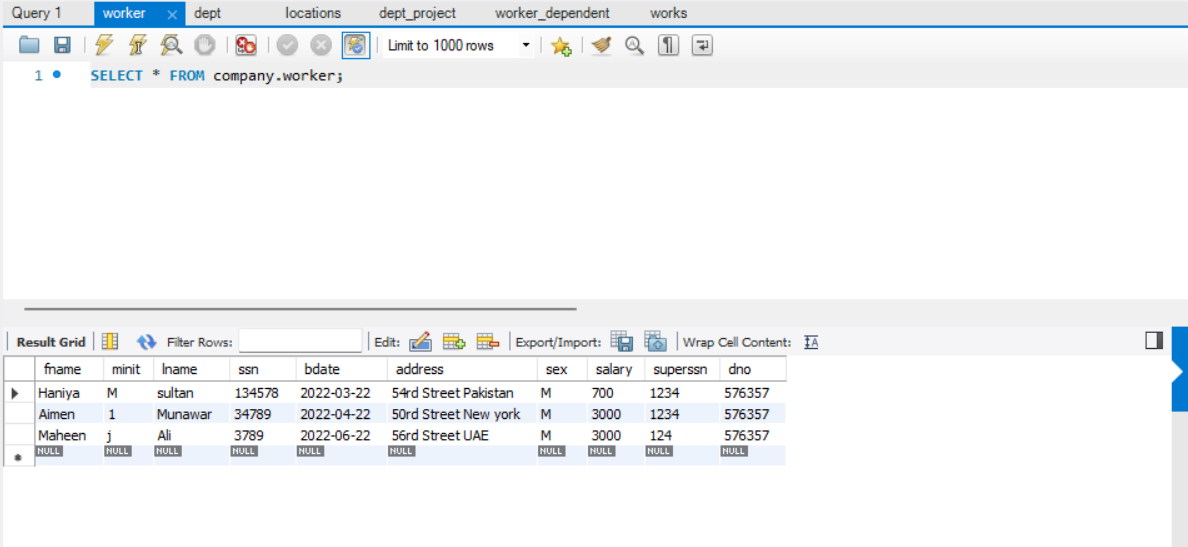
VALUES ('123456789', 1, 40);

INSERT INTO works (essn, pno, hours)

VALUES ('333445555', 2, 35);

INSERT INTO works (essn, pno, hours)

VALUES ('888665555', 3, 30);



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

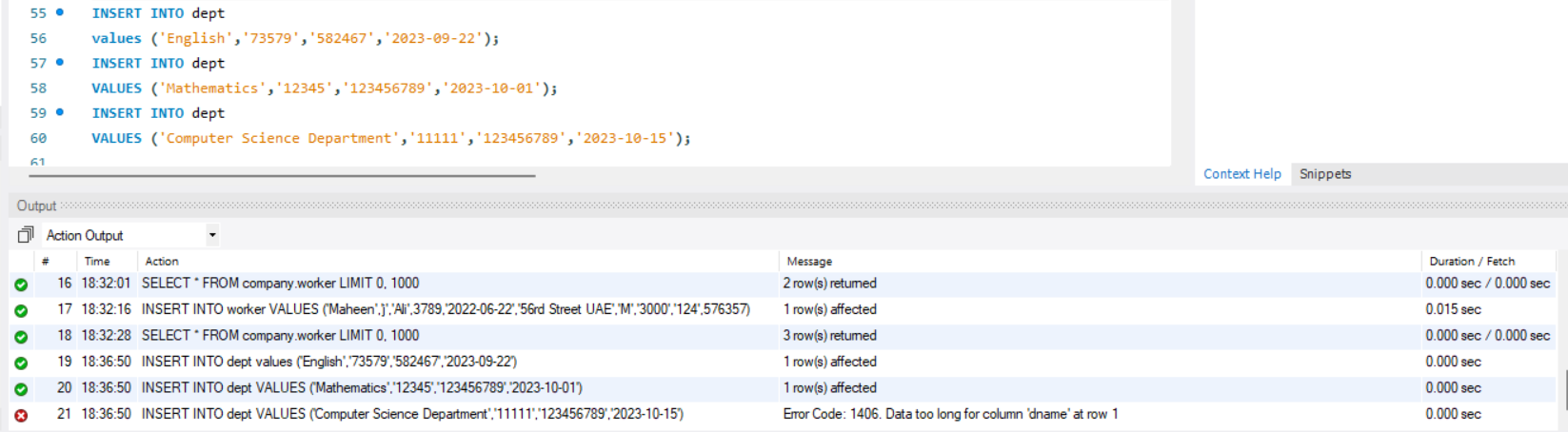
Description automatically generated

**Violations:**

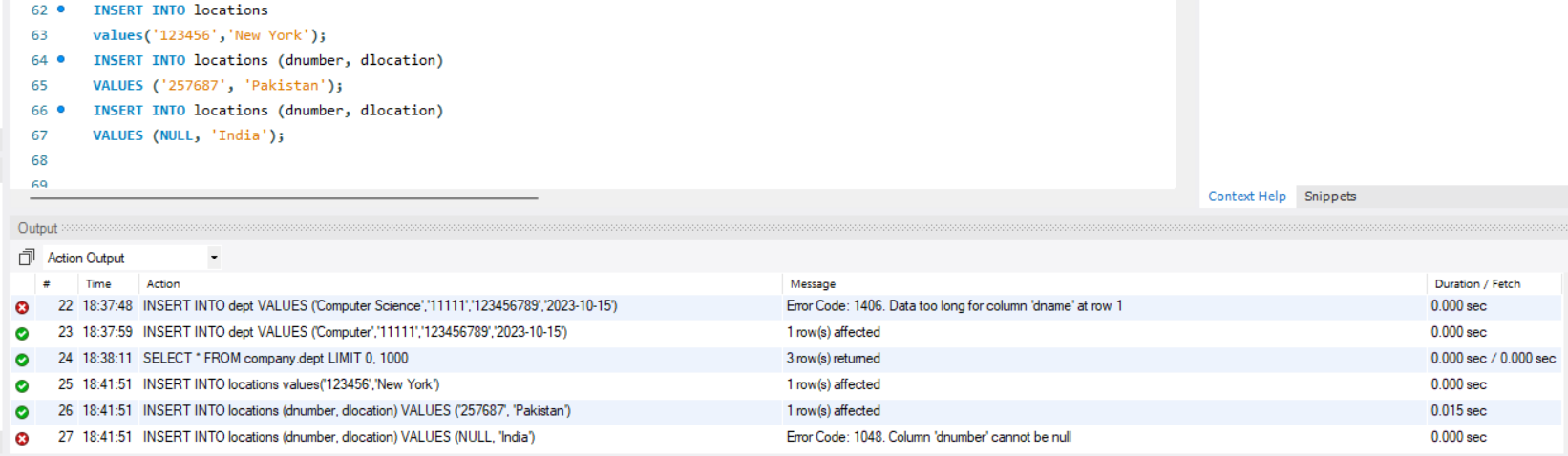
* Duplicate Value for Primary Key attribute



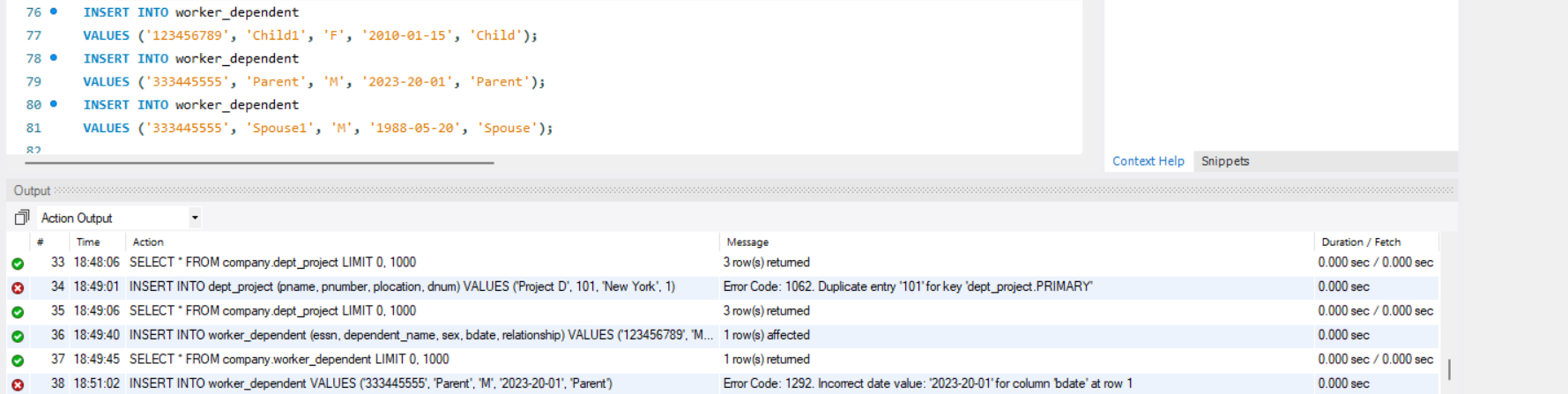
* Data too long



* Null entry for primary key attribute



* Incorrect Data Value



* 1. Try inserting some invalid data values and check how the DBMS responds. Record the error messages.
* **Error:** Data too long

A screenshot of a computer

Description automatically generated

* **Error:** Duplicate Value for Primary Key attribute

A screenshot of a computer

Description automatically generated

* **Error:** Null entry for primary key attribute

A screenshot of a computer

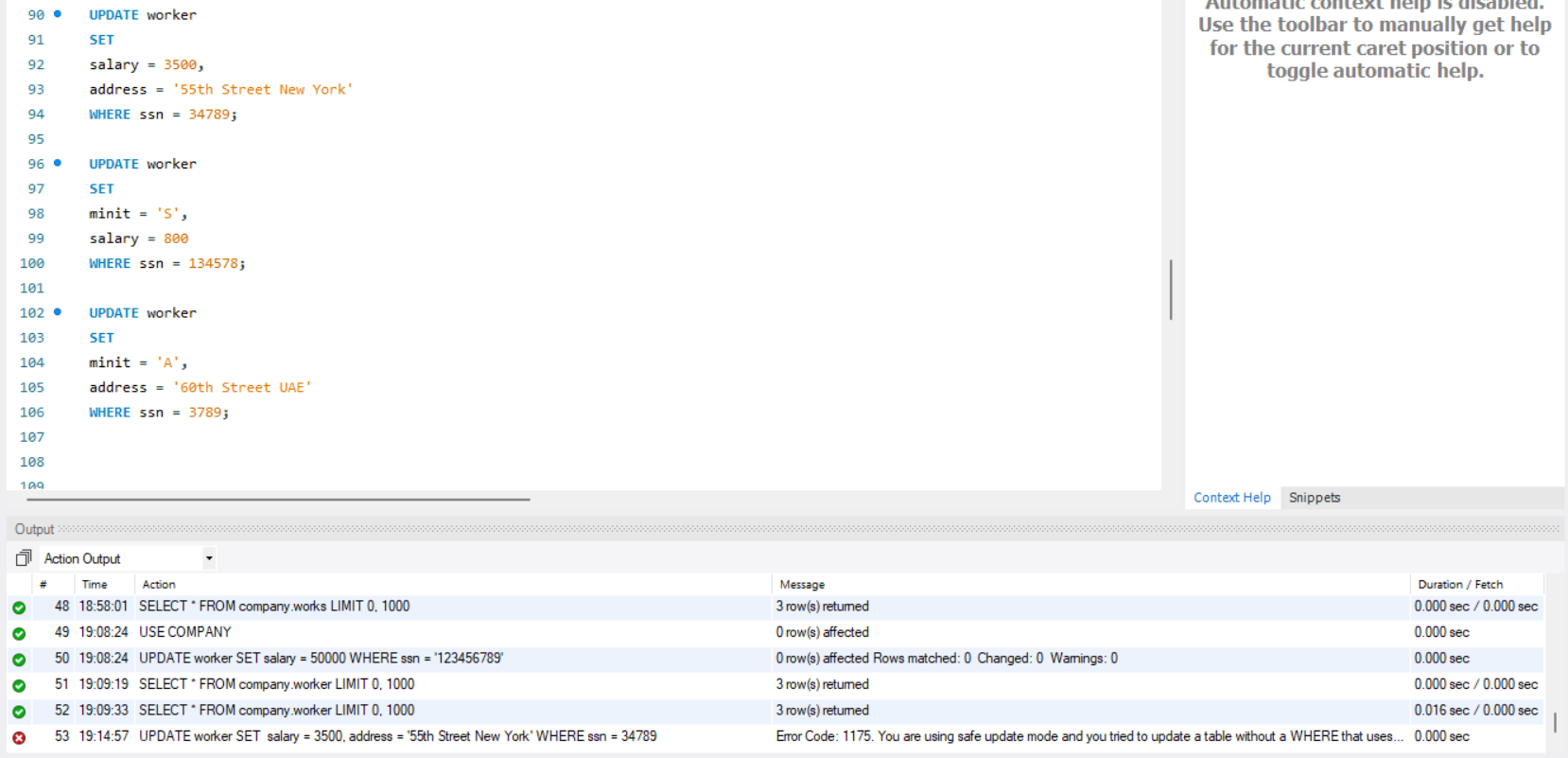
Description automatically generated

* **Error:** Incorrect Data Value

A screenshot of a computer

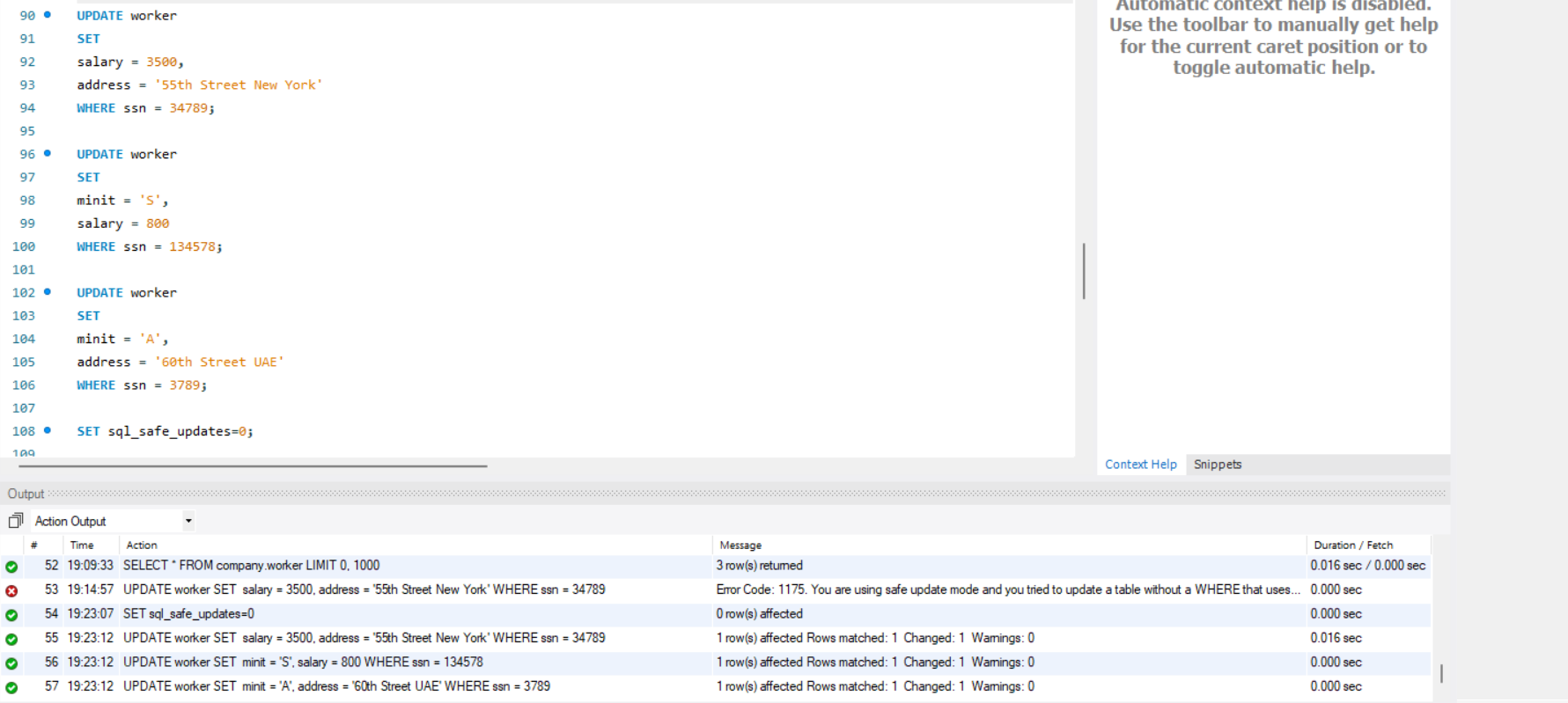
Description automatically generated

* 1. Update three of the rows using any column values; Observe whether a value is changed or not. Give your justifications.



The values in these records are not updated when we run these queries for the first time because we receive an error as the safe mode is turned ON. Since, in SQL, whenever we attempt to update a record with a primary key, it checks to see if the new value we are trying to put already exists in the column that constitutes the primary key. If it does and safe mode is toggled ON, the update won't take place. This is so that the uniqueness constraint wouldn't be broken by updating a primary key to a value that already exists.

After Turning the Safe mode OFF the records get updated successfully.



**Justifications:** The records will update if the safe mode is toggled OFF.

**Worker Table Before Update:**

A screenshot of a computer

Description automatically generated

**Worker Table After Update:**

A screenshot of a computer

Description automatically generated

* 1. Delete a record from the Dept\_Locations table. Can we delete a row from the table using the PK column? If yes how, if not why?

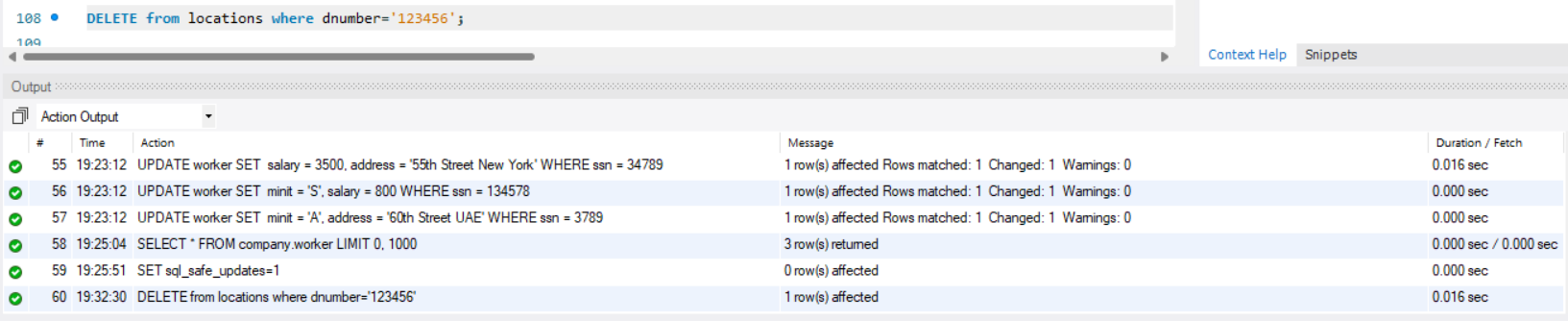
Yes, we can delete a record from a table using its PK using the **DELETE\_FROM\_WHERE** clause.

**Dept\_Location Table before Deletion:**



**Deleting Record with PK=123456:**

**CODE:** DELETE from locations where dnumber='123456';

****

**Dept\_Location Table after Deletion:**

**A white background with black dots

Description automatically generated**

1. Save your previous queries and then drop the Company Schema.

**CODE:**

drop table worker;

drop table worker\_dependent;

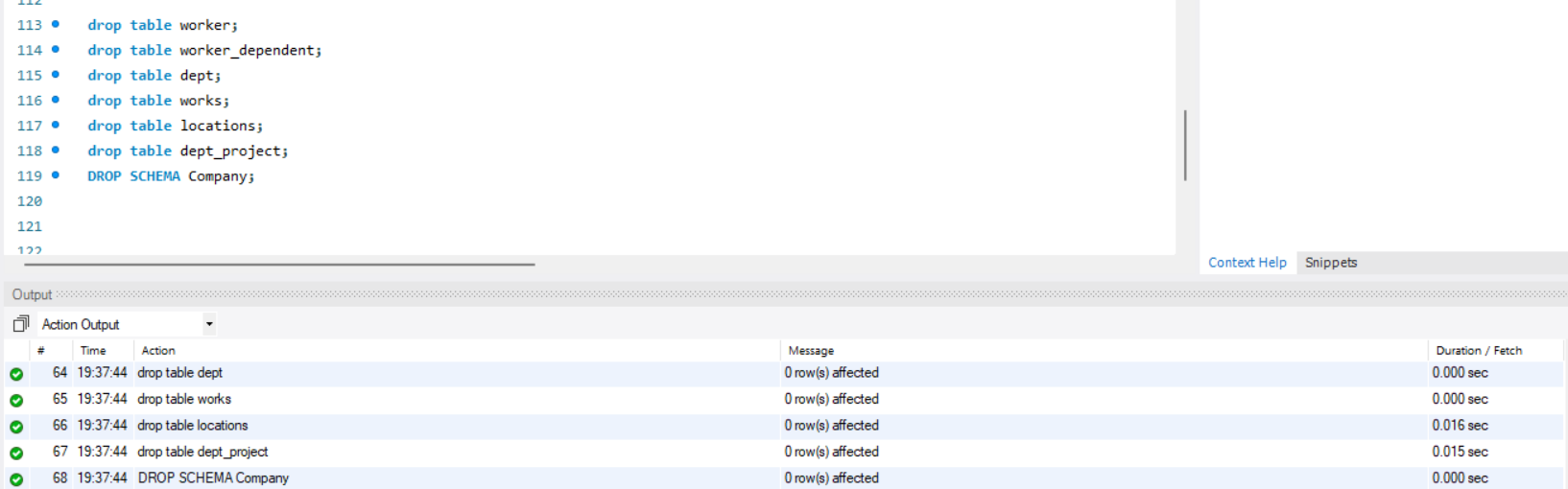
drop table dept;

drop table works;

drop table locations;

drop table dept\_project;

DROP SCHEMA Company;



1. Again create COMPANY database and the corresponding tables/relations in the Company schema using the DDL commands. For this time ensure all of the following constraints are specified during the table creation time:
   1. Primary Key
   2. Foreign Key
   3. Not Null
   4. Unique

Create FK, PK as shown in the figure. Bdate attribute should not be null while Dname should always be unique. Assign valid names to your constraints.

**CODE:**

CREATE database COMPANY;

USE company;

CREATE TABLE dept(

dname varchar(15) UNIQUE ,

dnumber INT ,

mgrssn char(9) not null,

mgrstartdate date,

constraint PK\_dept primary key (dnumber)

);

CREATE TABLE worker(

fname varchar(15),

minit varchar(1),

lname varchar(15),

ssn char(9) ,

bdate date not null,

address varchar(30),

sex char,

salary INT,

superssn char(9),

dno INT,

constraint PK\_worker primary key (ssn),

constraint FK\_worker foreign key (dno) references dept(dnumber)

);

CREATE TABLE locations (

dnumber INT ,

dlocation varchar(15) ,

constraint PK\_locations primary key (dnumber,dlocation),

constraint FK\_locations foreign key(dnumber) references dept(dnumber)

);

CREATE TABLE dept\_project (

pname varchar(15) not null ,

pnumber INT ,

plocation varchar(15),

dnum INT not null,

constraint PK\_dept\_project primary key (pnumber),

constraint FK\_dept\_project foreign key (dnum) references dept(dnumber)

);

CREATE TABLE works (

essn char(9) ,

pno INT ,

hours INT,

constraint PK\_works primary key (essn,pno),

constraint FK\_works\_1 foreign key(essn) references worker(ssn),

constraint FK\_works\_2 foreign key(pno) references dept\_project(pnumber)

);

CREATE TABLE worker\_dependent(

essn char(9) ,

dependent\_name varchar(15) ,

sex char,

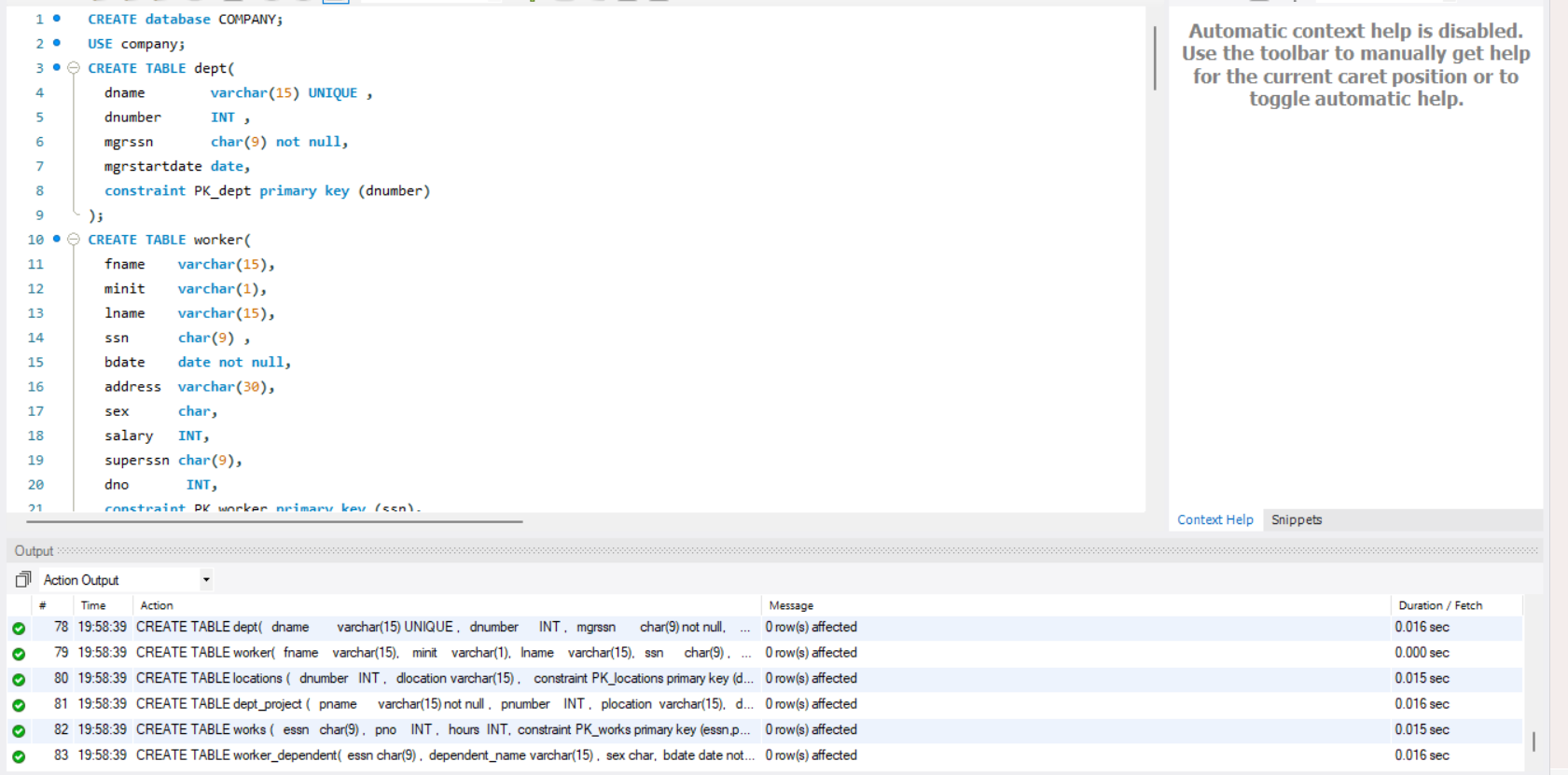
bdate date not null,

relationship varchar(20),

constraint PK\_worker\_dependent primary key (dependent\_name,essn),

constraint FK\_worker\_dependent foreign key(essn) references worker(ssn)

);



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**In order to specify an action in case of constraint violation, you will propagate the changes.**

1. Attempt question no. 2 again and answer the questions on the basis of new constraints added and their corresponding violations.
   1. Insert at least 3 rows of valid data into the tables. Follow the violations which occur while you insert data.

**Inserting Valid data:**

**CODE:**

INSERT INTO dept VALUES ('Finance', 1, '111223333', '2022-10-01');

INSERT INTO dept VALUES ('Marketing', 2, '222334444', '2022-09-15');

INSERT INTO dept VALUES ('IT', 3, '333445555', '2022-08-20');

INSERT INTO worker VALUES ('John', 'D', 'Doe', '111223333', '1990-05-20', '123 Main St', 'M', 60000, NULL, 1);

INSERT INTO worker VALUES ('Jane', 'A', 'Doe', '222334444', '1988-12-12', '456 Oak St', 'F', 55000, '111223333', 1);

INSERT INTO worker VALUES ('Michael', 'J', 'Smith', '333445555', '1995-03-30', '789 Pine St', 'M', 70000, '111223333', 2);

INSERT INTO locations VALUES (1, 'New York');

INSERT INTO locations VALUES (2, 'Los Angeles');

INSERT INTO locations VALUES (3, 'Chicago');

INSERT INTO dept\_project VALUES ('Project A', 101, 'New York', 1);

INSERT INTO dept\_project VALUES ('Project B', 102, 'Los Angeles', 2);

INSERT INTO dept\_project VALUES ('Project C', 103, 'Chicago', 3);

INSERT INTO works VALUES ('111223333', 101, 40);

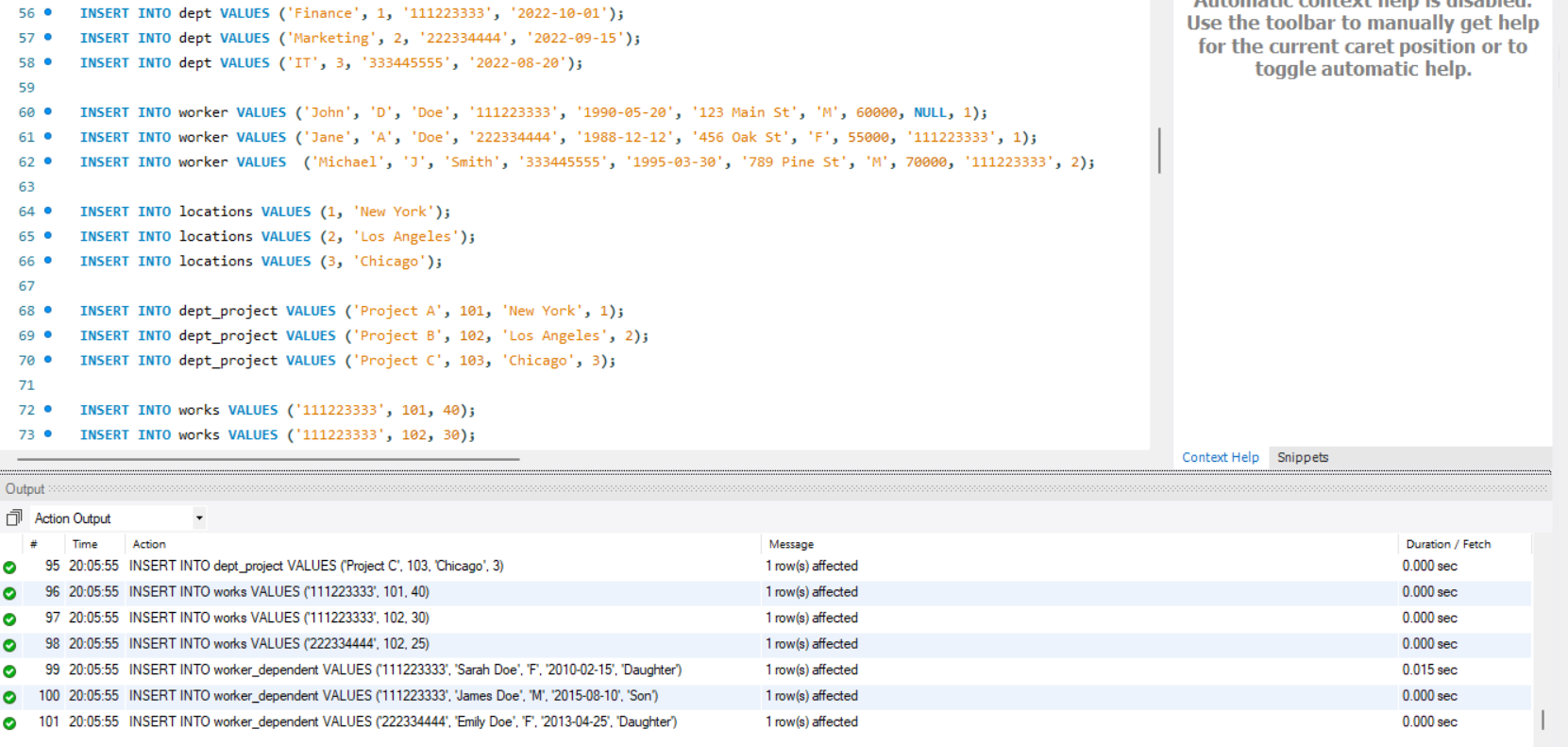
INSERT INTO works VALUES ('111223333', 102, 30);

INSERT INTO works VALUES ('222334444', 102, 25);

INSERT INTO worker\_dependent VALUES ('111223333', 'Sarah Doe', 'F', '2010-02-15', 'Daughter');

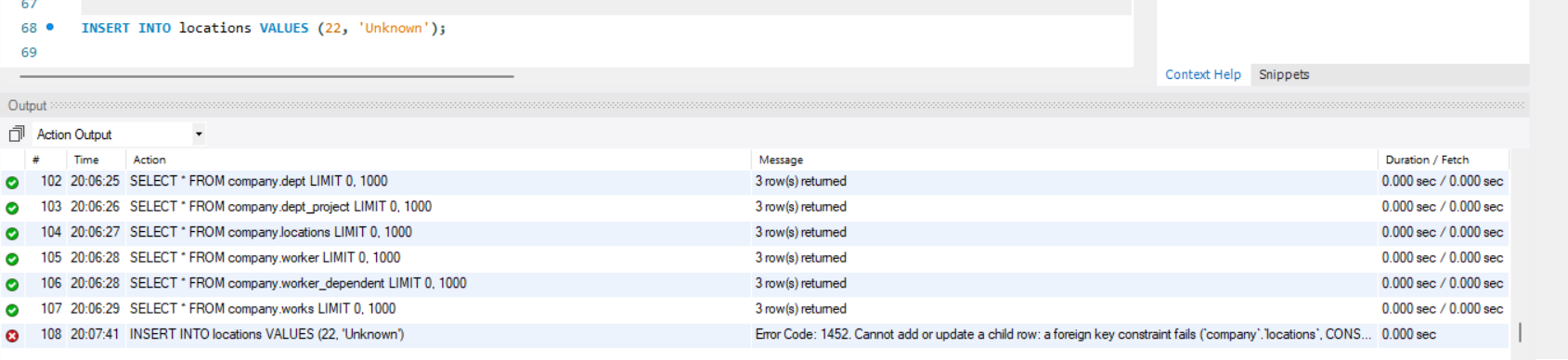
INSERT INTO worker\_dependent VALUES ('111223333', 'James Doe', 'M', '2015-08-10', 'Son');

INSERT INTO worker\_dependent VALUES ('222334444', 'Emily Doe', 'F', '2013-04-25', 'Daughter');



**Violations:**

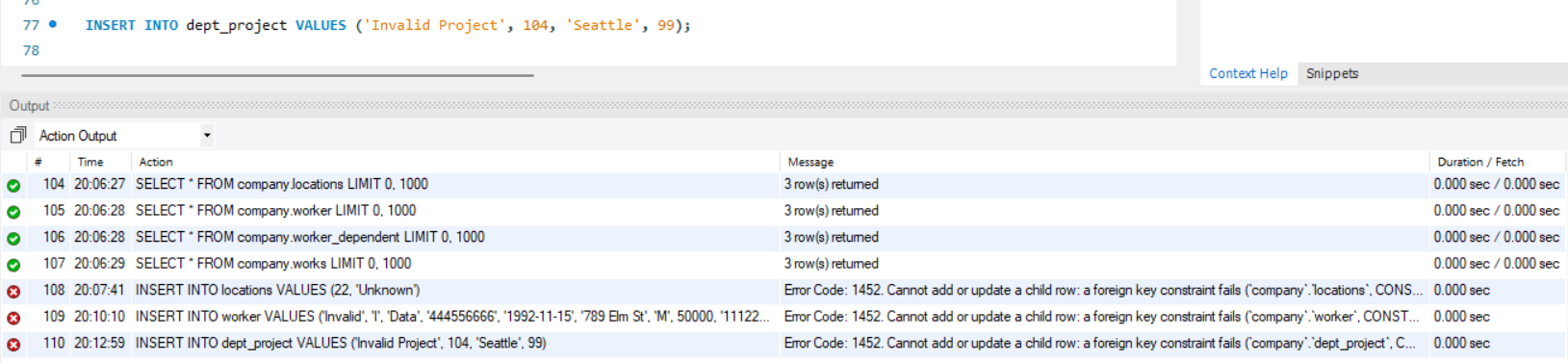
* **Foreign key constraint failed:** Trying to insert a location with a dept\_number that does not exist.



* 1. Try inserting some invalid data values and check how the DBMS responds. Record the error messages.

**Violation:** Trying to insert a project with a non-existent department number

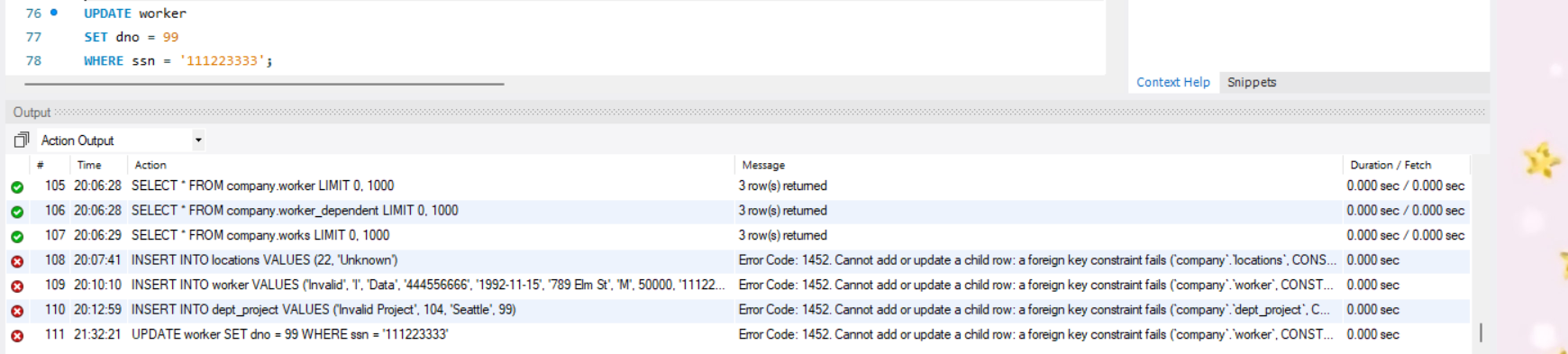
**Error:** Cannot add or update child row: a foreign key constraint fails.

****

* 1. Update three of the rows using any column values; Observe whether a value is changed or not. Give your justifications.

The update query will fail if we try to directly update a foreign key constraint in a child table without updating the corresponding value in the parent table.

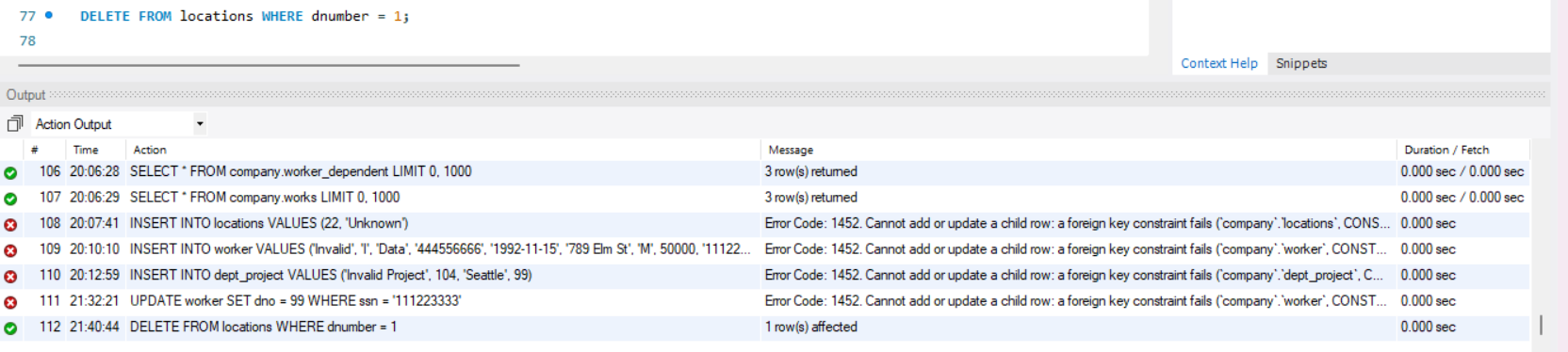
For example, if we try to update the dno in the worker table to a department number that doesn't exist in the dept table, it will result in a foreign key constraint violation and the query will fail.



* 1. Delete a record from the Dept\_Locations table. Can we delete a row from the table using the PK column? If yes how, if not why?

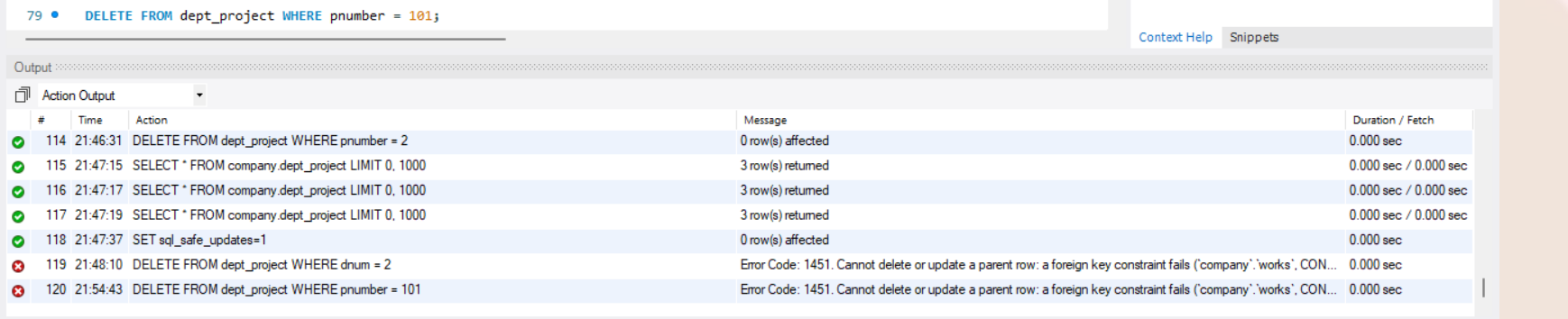
We can delete a row from the locations table using its primary key column (dnumber) because locations is the child table, and its foreign key references the primary key in the parent table (dept). The foreign key constraint only ensures that dnumber in locations always refers to an existing dnumber in dept.

**CODE:** DELETE FROM locations WHERE dnumber = 1;



In contrast, we encounter constraints if we try to delete a row from the dept\_project table using its primary key column (pnumber). This is because dept\_project is the parent table in this relationship.

**CODE:** DELETE FROM dept\_project WHERE pnumber = 101;

****