

Common Relational Database Operations

In this lesson, we will look at some of the different operations that can be performed on relational databases.

WE'LL COVER THE FOLLOWING ^

- The insert operation
- The delete operation
- The update operation

We will concentrate on the three basic operations that can change the states of relations in the database: Insert, Delete, and Update. **Insert** is used to insert one or more new tuples in a relation, **Delete** is used to delete tuples, and **Update** is used to change the values of some attributes in existing tuples.

Whenever these operations are applied, the integrity constraints specified on the relational database schema should not be violated. So we will also discuss the types of constraints that may be violated by each of these operations along with the types of actions that may be taken if an operation causes a violation.

We will be using the database state illustrated below:

EMPLOYEE

Name	Ssn	Bdate	Salary	Super_Ssn	Dept_Num
John Smith	333445555	1968-05-22	45,000	NULL	3
Emily Taylor	987654321	1972-09-01	30,000	333445555	3

Adam	66688444	1969-04-	55,000	33344555	3
Kovac	4	09		5	
Kevin	88866555	1979-09-	20,000	20394850	2
Jaimes	5	22		6	

DEPARTMENT

D_Name	D_No	Manager_Ssn
Administration	3	333445555
Research	2	678884823

The insert operation

The insert operation provides a list of attribute values for a new tuple t that is to be inserted into a relation R .

The insert operation can violate any of the four types of constraints:

- Domain constraints can be violated if an attribute value is given that does not appear in the corresponding domain or is not of the appropriate data type.
- Key constraints can be violated if the primary key value in the new tuple t already exists in another tuple in the relation.
- Entity integrity can be violated if any part of the primary key of the new tuple t is **NULL**.
- Referential integrity can be violated if the value of any foreign key in t refers to a tuple that does not exist in the referenced relation.

The following slides include examples of the insert operation:

Operation:

Insert <'James Willems', '334599005', '1970-01-29' , 75000, '333445555', 3> into EMPLOYEE

Result: This insertion is successful. So the table will be updated as follows:

Name	Ssn	Bdate	Salary	Super_Ssn	Dept_Num
John Smith	333445555	1968-05-22	45,000	NULL	3
Emily Taylor	987654321	1972-09-01	30,000	333445555	3
Adam Kovac	666884444	1969-04-09	55,000	333445555	3
Kevin Jaimes	888665555	1979-09-22	20,000	203948506	2
James Willems	334599005	1970-01-29	75,000	333445555	3

Table is updated with the new record.

1 of 3

Operation:

Insert <Jack DylanâNULL, â1960-06-15â18000, NULL, 2> into EMPLOYEE.

Result: This insertion violates the entity integrity constraint as there is a NULL value for the primary key Ssn. Thus, this insertion will be rejected.

Name	Ssn	Bdate	Salary	Super_Ssn	Dept_Num
John Smith	333445555	1968-05-22	45,000	NULL	3
Emily Taylor	987654321	1972-09-01	30,000	333445555	3
Adam Kovac	666884444	1969-04-09	55,000	333445555	3
Kevin Jaimes	888665555	1979-09-22	20,000	203948506	2

Table remains the same.

2 of 3

Operation:

Insert <Bruce Green, 987654321, 1960-10-02, 30000, 333445555> into EMPLOYEE.

Result: This insertion violates the referential integrity constraint specified on Dept_Num in EMPLOYEE because no corresponding referenced tuple exists in DEPARTMENT with D_No equal to 5.

So, again insertion will be rejected.

Name	Ssn	Bdate	Salary	Super_Ssn	Dept_Num
John Smith	333445555	1968-05-22	45,000	NULL	3
Emily Taylor	987654321	1972-09-01	30,000	333445555	3
Adam Kovac	666884444	1969-04-09	55,000	333445555	3
Kevin Jaimes	888665555	1979-09-22	20,000	203948506	2

Table remains the same.

3 of 3

—

[]

If an insertion violates one or more constraints, the default option is to **reject** the insertion. In this case, it would be useful if the DBMS could provide a reason as to why the insertion was rejected.

The delete operation

The delete operation can violate only referential integrity. This occurs if the tuple being deleted is referenced by foreign keys from other tuples in the database. To specify deletion, a condition of the attributes of the relation selects the tuple (or tuples) to be deleted.

Here are a few examples of the delete operation:

Operation:

Delete the EMPLOYEE tuple with `Ssn` = '666884444'.

Result: This deletion is valid and the table will be updated as follows:

Name	Ssn	Bdate	Salary	Super_Ssn	Dept_Num
John Smith	333445555	1968-05-22	45,000	NULL	3
Emily Taylor	987654321	1972-09-01	30,000	333445555	3
Kevin Jaimes	888665555	1979-09-22	20,000	203948506	2

Table is updated with the deleted record.

1 of 2

2. Operation:

Delete the DEPARTMENT tuple with D_No = 3.

Result: This deletion is not acceptable, because there are tuples in EMPLOYEE relation that refer to this tuple via the Dept_Num foreign key. Hence, if this tuple is deleted, referential integrity will be violated.

D_Name	D_No	Manager_Ssn
Administration	3	333445555
Research	2	678884823

The DEPARTMENT table remains the same.

2 of 2

Several options are available if a deletion operation causes a violation. The first option, called **restrict**, is to reject the deletion. The second option, called **cascade**, is to attempt to cascade the deletion by deleting tuples that reference the tuple that is being deleted.

The update operation

The update operation is used to change the values of one or more attributes in a tuple (or tuples) of some relation R . It is necessary to specify a condition on the attributes of the relation to select the tuple (or tuples) to be modified.

Here are a few examples of the update operation:

Operation:

Update the 'Salary' of the EMPLOYEE tuple with Ssn = '888665555' to 45

Result: this update operation is acceptable, and the table will change as follows:

Name	Ssn	Bdate	Salary	Super_Ssn	Dept_Num
John Smith	333445555	1968-05-22	45,000	NULL	3
Emily Taylor	987654321	1972-09-01	30,000	333445555	3
Adam Kovac	666884444	1969-04-09	55,000	333445555	3
Kevin Jaimes	888665555	1979-09-22	45,000	203948506	2

The appropriate record has been updated.

Operation:

Update the Dept_Num of the EMPLOYEE tuple with Ssn = 888665555 to 5.

Result: Unacceptable, because it violates referential integrity.

Name	Ssn	Bdate	Salary	Super_Ssn	Dept_Num
John Smith	333445555	1968-05-22	45,000	NULL	3
Emily Taylor	987654321	1972-09-01	30,000	333445555	3
Adam Kovac	666884444	1969-04-09	55,000	333445555	3
Kevin Jaimes	888665555	1979-09-22	20,000	203948506	2

The table remains the same.

2 of 3

Operation:

Update the Ssn of the EMPLOYEE tuple with Ssn = 888665555 to 98765

Result: Unacceptable, because it violates primary key constraint by repeating a value that already exists as a primary key in another tuple.

Name	Ssn	Bdate	Salary	Super_Ssn	Dept_Num
John Smith	333445555	1968-05-22	45,000	NULL	3
Emily Taylor	987654321	1972-09-01	30,000	333445555	3
Adam Kovac	666884444	1969-04-09	55,000	333445555	3
Kevin Jaimes	888665555	1979-09-22	20,000	203948506	2

The table remains the same.

3 of 3

Updating an attribute that is neither part of a primary key nor part of a foreign key usually causes no problems; the DBMS need only check to confirm that the new value is of the correct data type and domain.

Modifying a primary key value is similar to deleting one tuple and inserting another in its place, because we use the primary key to identify tuples. Hence, the issues discussed earlier in insertion and deletion operations come into play.

The next lesson includes a quiz to test your knowledge regarding relational databases.