Referential Integrity

Objectives

Referential Integrity

Db2 provides many facilities to safeguard the integrity of the data it holds.

One of these is referential integrity (RI) which makes use of the primary and foreign keys within tables. Another facility is check constraint, which constrains data within a row.

After completing this module you will be able to:

- Define and Code Referential Integrity
- Define and Code Check Constraints

EMP_ID	SALARY	DEPT_ID
223344	36000.00	D123
166211	72000.00	D999
711792	28000.00	D123

Employee

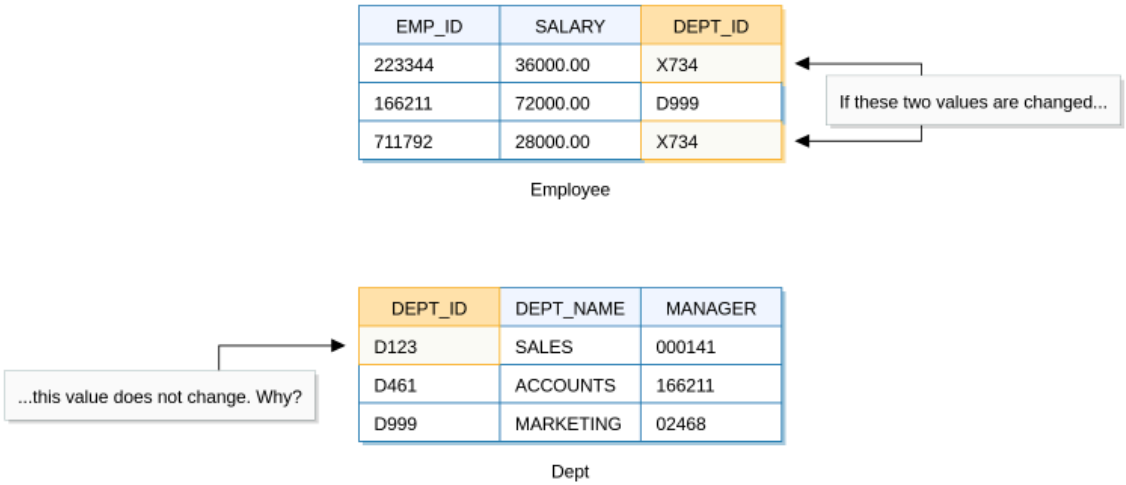
DEPT_ID	DEPT_NAME	MANAGER
D123	SALES	000141
D461	ACCOUNTS	166211
D999	MARKETING	02468

Dept

Referential integrity (RI) is concerned with keeping a database consistent and usable, especially after updates and deletes have taken place.

Take a look at these two tables.

If the D123 (Dept_ID) is changed in the Employee table, consider what should happen to the D123 (Dept_ID) in the Dept table.



Without referential integrity there would be employees in a fictitious department after the update.

To make the data consistent again you would need to change the value in the Dept table or add a new department.

Db2 has built-in facilities to make RI a little easier.

EMP_ID	SALARY	DEPT_ID
223344	36000.00	D123
166211	72000.00	D999
711792	28000.00	D123

Foreign key

Employee

Primary key

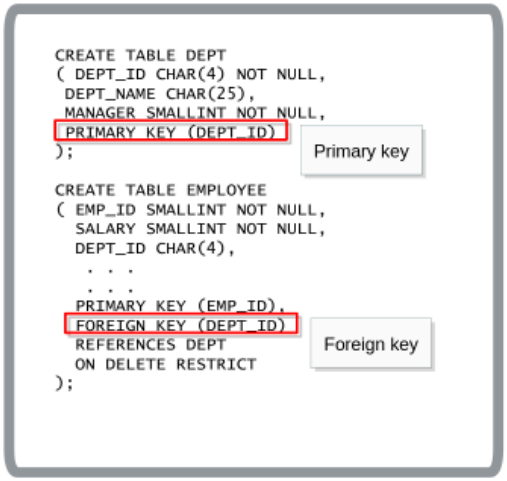
DEPT_ID	DEPT_NAME	MANAGER
D123	SALES	000141
D461	ACCOUNTS	166211
D999	MARKETING	02468

Dept

Db2 makes use of primary and foreign keys to maintain referential integrity. These are defined when the table is created or added later by using an ALTER statement.

The decision to use Db2 RI facilities is usually made when the database is designed. A programmer must know whether the facilities are being used. If there is no Db2 RI, it is up to the programmer to maintain referential integrity.

If Db2 RI is being used, depending on the options chosen, some things are done for you automatically.



The primary key of a table must be unique; hence, the NOT NULL on column definitions.

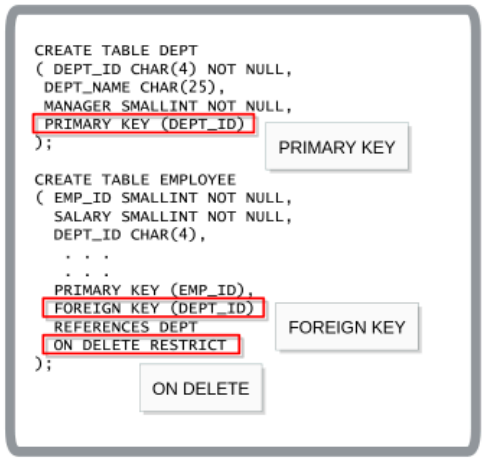
A table can only have one primary key but may have many foreign keys.

All primary keys and unique keys must have a unique index created for them.

The data types and lengths of the columns that make up the primary key and the foreign key must match.

If you are creating the Dept and Employee tables and you want to use Db2 RI facilities, you must tell Db2 what the primary and foreign keys are for each table.

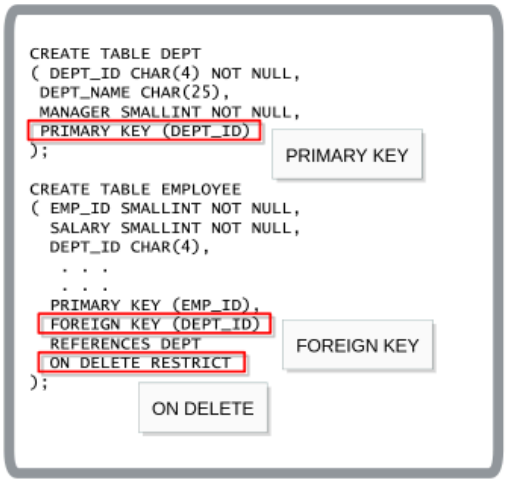
These two CREATE statements show the definition of primary and foreign keys.



In this example the PRIMARY KEY clause in the CREATE statement for the Dept table tells Db2 this column can be used for RI checking.

The FOREIGN KEY statement shown in the EMPLOYEE table is used to identify the column in the Employee table to be assigned as a foreign key. The REFERENCES DEPT code under this tells Db2 that the DEPT table holds the unique or primary key that the foreign key is to reference. In this scenario nothing follows the REFERENCES DEPT code so Db2 assumes that the primary key from the DEPT table is the reference. If another column needed to be used as reference, its name would appear in parenthesis after the REFERENCES DEPT code.

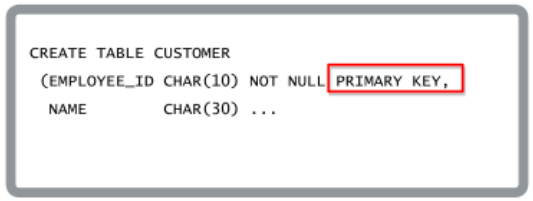
The ON DELETE tells Db2 what action to take if an attempt is made to delete a primary key that has dependent foreign keys.

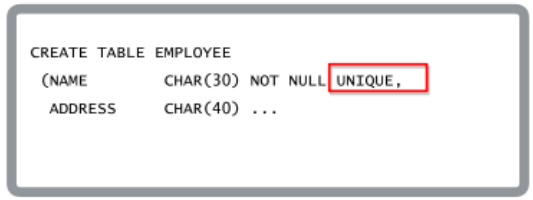


The options for ON DELETE are:

- RESTRICT indicates that a row in the parent table cannot be deleted if there are dependent rows.
- SET NULL indicates that if a parent table primary key is deleted, the foreign keys in any dependent rows will be set to NULL.
- CASCADE indicates that if the primary key in the parent table is deleted, then the foreign key that references it will also be deleted.
- NO ACTION is the default setting and is similar to RESTRICT.

The options for ON DELETE are shown in this table.





On previous pages you have seen how a primary key is defined for a table as a unique constraint, appearing as code after column definitions. Another method used for defining single columns as a primary or unique key, is shown here. In this case, the code is part of the column definition.

INSERT

A value inserted into the foreign key must match a value of the associated key in the parent table otherwise the insert action is disallowed.

A new row can be inserted into the table with the primary key, as long as the value is unique.

UPDATE

A value updated in the foreign key must match a value of the associated key in the parent table.

A value that is being updated in a primary key must not match any existing foreign key value.

Db2 will enforce certain referential integrity rules when a user performs an action on a table containing a primary, unique or foreign key. Here are some insert and update rules that you should be aware of (delete rules have been discussed on previous pages).

Advantages	Disadvantages
DB2 automatically keeps data relationally consistent. You do not have to code extra record checks and deletes. Foreign or primary key relationships document and secure the relationship between tables in a database.	RI forces you to do some things in a particular order. For example, you must insert primary key rows first. Erroneous deleting can have unexpected consequences both for data and performance. Updates, deletes, and inserts require more processing as DB2 checks foreign keys.

Usually some form of RI is implemented when a system is built, but with Db2 you have the choice of doing it all yourself or letting Db2 do some of the work.

Db2 RI has both advantages and disadvantages as shown here.

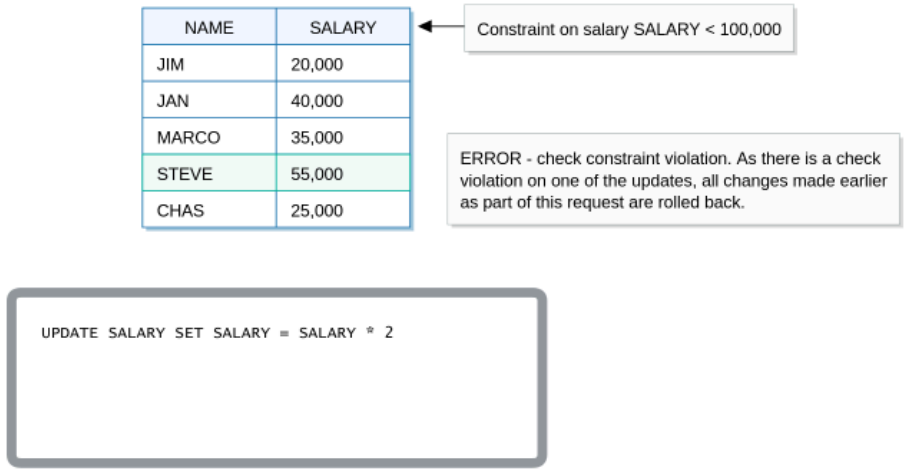
- NOT NULL
- COLUMN DEFAULTS
- UNIQUE CONSTRAINTS
- CHECK CONSTRAINTS
- PRIMARY KEY CONSTRAINTS
- FOREIGN KEY CONSTRAINTS

If no data is provided for the column, it is populated with the default value.

So far this module has explored how Db2 provides the above constraints on the value of data in a column or table. Now you will look at check constraints.

Check constraints provide a method for ensuring the validity of data in a table while removing the responsibility from the application program.

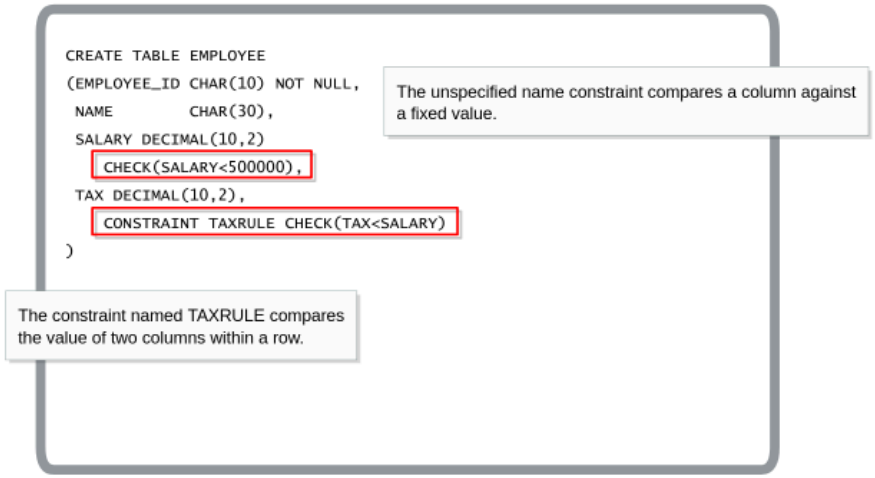
Mouse-over the list for a summary of the constraints.



A check constraint contains a condition, which must not be evaluated as false when an action such as an insert or load is performed. If the check condition does not meet these requirements, then a check violation occurs and the action is not performed.

For example, you could have a condition that checks that the value of the SALARY column in a table is between 10000 and 100000. If an attempt is made to insert a record where the SALARY value is 200000 then the insert would be disallowed.

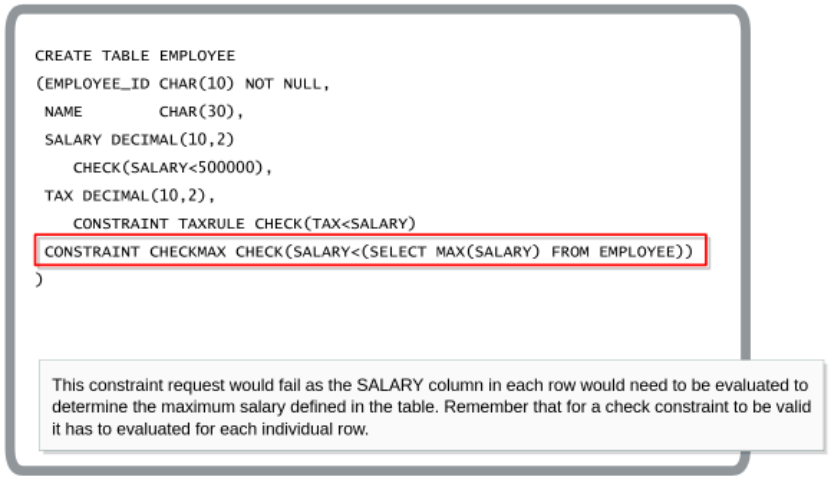
Click **Play** to see what occurs when an entire table update is requested.



You can specify constraints for columns when creating a table or they can be added later using the ALTER TABLE statement.

Db2 will check the constraints when data is updated, inserted, or loaded into a table.

The CHECK statement is used to identify a check constraint to Db2. An optional constraint name can be applied to the check constraint by specifying CONSTRAINT name. If a constraint name is not specified then Db2 will automatically generate one.



A constraint must be able to be evaluated by examining a single row. It can be any valid Db2 predicate that references the current row only or multiple predicates joined by AND and OR.

Some examples are:

column IN (A,B,C,D)
column < 100
column between 100 and 200
column1 > column2 AND column1 < column3

Summary

Referential Integrity

In this module you have examined referential integrity and the use of primary and foreign keys within tables. You have also looked at check constraints which constrain data within a row.

You now should be able to:

- Define and Code Referential Integrity
- Define and Code Check Constraints