

Database Constraints

Database Constraints

- Database constraints are restrictions on the contents of the database or on database operations
- Database constraints provide a way to guarantee that:
 - rows in a table have valid primary or unique key values
 - rows in a dependent table have valid foreign key values that reference rows in a parent table
 - individual column values are valid

Database Constraints

- SQL/400 constraints cover four specific integrity rules:
 - primary key constraint (to enforce existence integrity)
 - unique constraint (to enforce candidate key integrity)
 - foreign key constraint (to enforce foreign key, or referential integrity)
 - check constraint (to restrict a column's values; a partial enforcement of domain integrity)
- You specify one or more of these constraints when you use the Create Table statement to create a base table.
- You can also add or drop constraints with the Alter Table statement.

Database Constraints

- UDB/400 enforces all four types of constraints when rows in the table are inserted or updated
- In the case of a foreign key constraint → when rows in the parent table are updated or deleted
- Once a constraint for a table is defined
 - The constraint is enforced for *all* database updates through *any* interface
 - including SQL DML, HLL built-in I/O operations, Interactive SQL (ISQL) ad hoc updates, etc.

Create Table Sale

(OrderID Dec(7, 0) Not Null
Constraint SaleOrderIdChk Check(OrderID > 0),
 SaleDate Date Not Null,
 ShipDate Date Default Null,
 SaleTot Dec(7, 2) Not Null,
 CrdAutNbr Int Default Null,
 CustID Dec(7, 0) Not Null,

Primary Key(OrderID),

Constraint SaleCrdAutNbrUK Unique (CrdAutNbr),
Constraint SaleCustomerFK Foreign Key (CustID)
References Customer (CustID)

On Delete Cascade

On Update Restrict,

Constraint SaleShipDateChk Check(ShipDate Is Null

Or ShipDate >= SaleDate))

Database Constraints

- If you don't specify a constraint name, UDB/400 generates a name when the table is created.
 - The constraint name can later be used in the Alter Table statement to drop the constraint

Primary Key Constraints

- A primary key serves as the unique identifier for rows in the table.
- The syntax of the **primary key constraint** (following the Constraint keyword and the constraint name, if they're specified) is
 - Primary Key(*column-name*, ...)
 - Each primary key column's definition must include Not Null.
 - For a table with a primary key constraint, UDB/400 blocks any attempt to insert or update a row that would cause two rows in the same table to have identical value(s) for their primary key column(s).
 - A table definition can have no more than one primary key constraint.

Unique Constraints

- A **unique constraint** is similar to a primary key constraint → doesn't have to be defined with Not Null.
- Recommend → always specify a constraint name for a unique constraint:
 - Constraint *constraint-name* Unique (*column-name*, ...)
- Note that a unique constraint does *not* use the Key keyword, as do primary key and foreign key constraints.
- The **SaleCrdAutNbrUK** unique constraint specifies that any *non-null* value for the CrdAutNbr column must be unique.
- Allowing the CrdAutNbr column to be null and specifying the SaleCrdAutNbrUK constraint together enforce a business rule that some orders (e.g., those paid by cash) may exist without a credit authorization number, but any order that does have a credit authorization number must have a unique value.

Foreign Key Constraints

- A **foreign key constraint** specifies how records in different tables are related and how UDB/400 should handle row insert, delete, and update operations that might violate the relationship.
- For example, sales rows are generally related to the customers who place the orders. Although it might be valid for a customer row to exist without any corresponding sale rows, it would normally be invalid for a sale row not to have a reference to a valid customer.
- With a relational DBMS, the relationship between rows in two tables is expressed by a **foreign key** in the **dependent table**. A foreign key is one or more columns that contain a value identical to a primary key (or unique key) value in some row in the **parent table** (i.e., the **referenced table**).

- With SQL/400, we might create the Customer and Sale tables so they have the following partial constraint definitions:
 - Customer table (parent)
 - Primary key column: CustID
 - Sale table (dependent)
 - Primary key column: OrderID
 - Foreign key column: CustID
- For each row in the Sale table, the CustID column should contain the same value as the CustID column of some Customer row because this value tells which customer placed the order.
- The purpose of specifying a foreign key constraint is to have UDB/400 ensure that the Sale table never has a row with a (non-null) value in the CustID column that has no matching Customer row.

Foreign Key Constraints cont.

The Sale table's foreign key constraint, which is

Constraint SaleCustomerFK Foreign Key (CustID)
References Customer (CustID)
On Delete Cascade
On Update Restrict

- Specifies that the CustID column in the Sale table is a foreign key that references the CustID primary key column in the Customer table.
- UDB/400 does not allow an application to insert a new row in the Sale table unless the row's CustID column contains the value of some existing CustID value in the Customer table.
- Blocks any attempt to change the CustID column of a row in the Sale table to a value that doesn't exist in any row in the Customer table. [a new or updated Sale row must have a parent Customer row].

A foreign key constraint can specify the same table for the dependent and parent tables. Suppose you have an Employee table with an EmpID primary key column and a MgrEmpID column that holds the employee ID for the person's manager. :

Create Table Employee

```
( EmpID      Dec ( 7, 0 ) Not Null,  
  MgrEmpID   Dec ( 7, 0 ) Not Null,  
  other column definitions ... ,
```

```
Primary Key ( EmpID ),
```

```
Constraint EmpMgrFK Foreign Key ( MgrEmpID )
```

```
References Employee      ( EmpID )
```

```
On Update Restrict
```

```
On Delete Restrict )
```

Check Constraints

- Used to enforce the validity of column values.
 - Constraint SaleOrderIdChk Check(OrderID > 0)
[guarantees that the OrderID primary key column is always greater than zero]
 - Constraint SaleShipDateChk Check(ShipDate Is Null Or ShipDate >= SaleDate)
[guarantees that either a row has no ship date (i.e., the ShipDate column is null, meaning "unknown") or the ship date is on or after the sale date].
- A check constraint can compare a column to a constant (such as in the first example), to another column in the same table (such as in the second example), or to an expression (e.g., ColA + 3).
- UDB/400 checks to make sure a new or changed row doesn't violate any of its table's check constraints before an insert or update operation is allowed.

check constraints cont.

You can combine check constraints for more than one column into a single check constraint, as in the following example:

Constraint CustStatusNameChk

```
Check ( ( Status = 'A' Or Status = 'I' )  
And ( Name <> '' ) )
```

Add/Remove Constraints

- After you create a table, you can use the Alter Table statement to
 - add or remove a primary key, unique, foreign key, or check constraint
- To drop a table's primary key constraint, just specify the Primary Key keywords:

**Alter Table Sale
Drop Primary Key**

- To drop a unique, foreign key, or check constraint, you must specify the constraint name:

**Alter Table Sale
Drop Constraint SaleCustomerFK**

- To add a new constraint, use the same constraint syntax as in a Create Table statement:

**Alter Table Sale
Add Constraint SaleSaleTotChk Check(SaleTot >= 0)**

Database Constraints

- Constraints can be added to a file in several ways
 - the CL Add Physical File Constraint (ADDPFCST) command,
 - the Create Table or Alter Table SQL statements,
 - the Table Properties tab of AS/400 Operations Navigator.
- Because constraints are defined at the file level, adding one to an existing file requires that all existing data comply with the constraint being added. You might need to perform data cleanup prior to introducing a constraint to the file.
- If you try adding a constraint to a file already containing data that violates the constraint, the constraint is added in a check pending status.
 - A constraint having a status of check pending doesn't become enabled until after the data in violation of the constraint is corrected and the constraint readded.
 - To identify and correct the records in violation, two CL commands exist to assist you-Work with Physical File Constraints (WRKPFCST) and Display Check Pending Constraint (DSPCPCST).

Constraint States

- Constraints added to a file can have a state of either *enabled* or *disabled*.
- When set to disabled, the rules of the constraint are not enforced. If a constraint is added in a check pending status, the state is automatically set to disabled.
 - An active constraint can be temporarily disabled using the State parameter of the Change Physical File Constraint (CHGPFCST) command.

Database Constraints

- Database constraints provide a way to guarantee that:
 - rows in a table have valid primary or unique key values
 - rows in a dependent table have valid foreign key values that reference rows in a parent table
 - individual column values are valid

Database Constraints

- Database constraints provide a way to guarantee that:
 - rows in a table have valid primary or unique key values
 - rows in a dependent table have valid foreign key values that reference rows in a parent table
 - individual column values are valid

Section A - (24 x 90)

File Edit Insert Appearance Communication Agent Window Help

Print Screen Copy Paste Send Recv Display Color Map Record Stop Ctrl Play Ctrl Out

Work with Physical File Constraints

Type options, press Enter.
 2*Change 4*Remove 6*Display records in check pending

Opt	Constraint	File	Library	Type	State	Check Pending
6	EMP_EMP_NU	EMPLOYEE	MARKTEST	*CHKCST	EST/DSB	YES

Bottom

Parameters for options 2, 4, 6 or command
 ***)

F3*Exit F4*Prompt F5*Refresh F9*Retrieve F12*Cancel F15*Sort by
 F16*Repeat position to F17*Position to F22*Display constraint name

YES a NU

Connected to remote server/host s101901r using port 23