Constraints

Constraints

- Sometimes referred to as integrity constraints or integrity rules
- Restrictions based on business rules and other business policies and procedures that ensure data in a database are acceptable and accurate

Constraint Categories

- Data integrity
- Entity integrity
- Referential integrity

Data Integrity

- Data integrity defines the possible values of a column
- In a database system, data integrity is defined by:
  - Data type and length
  - NULL value acceptance
  - Allowable values
  - Default value

Entity Integrity

- Every tables must have a primary key
- Candidate keys - More than one possible set of columns that may meet the criteria for a primary key
- Composite primary keys – a primary key that is composed of more than one column
- Primary key cannot contain NULL

Referential Integrity

- Concept that ensures that relationships between tables remain reliable
- Foreign keys
- The concept of referential integrity states that a row containing the foreign key may not be added to the table unless a matching value exists in the primary key column of the parent table

Constraint Types

Constraint	Description
PRIMARY KEY	Identifies which column is the unique identifier or primary key for each row in the table. The values in the primary key column must be unique for every row in the table. Since the primary key must be unique, it cannot be NULL.
FOREIGN KEY	Is what makes the relational database work. For every one-to-many (parent-child) relationship in the database, a foreign key constraint is added to the child (many) table. The foreign key in the child (many) table links to the parent (one) table. Thus, if a row is added to the child table, the value entered in the foreign key column must already exist as a primary key in the parent table.
UNIQUE	Identifies a column as containing unique values for the UNIQUE column in each row in the table. The UNIQUE constraint differs from a primary key in that it allows NULL values.
CHECK	Enforces a business rule on a column. Before a value can be entered into a CHECK column, the condition (business rule) specified in the CHECK constraint must be true.
NOT NULL	Ensures that a column identified as NOT NULL will not contain a NULL value.

Primary Key Constraint

- Each row in a table contains one or more columns that uniquely identify that row in the table

```
CREATE TABLE customers
(  customer_id   INTEGER                NOT NULL,
   customer_name CHARACTER( 30    )     NOT NULL,
   balance       DECIMAL  (  7, 2 )     NOT NULL DEFAULT 0,
   ship_city     CHARACTER( 30    )     NOT NULL,
   credit_limit  DECIMAL  (  7, 0 )     NOT NULL DEFAULT 100000,
   discount      DECIMAL  (  5, 3 )
  CONSTRAINT customers_customer_id_pk  PRIMARY KEY(customer_id) );
```

```
  ALTER TABLE customers
    ADD CONSTRAINT customers_customer_id_pk
        PRIMARY KEY(customer_id);
```

Defining the Primary Key Constraint

- Identified with the keyword CONSTRAINT
- Followed by constraint name, customers_customer_id_pk
- The constraint name is composed of three components separated by underscores

Defining the Primary Key Constraint

```
CONSTRAINT customers_customer_id_pk  PRIMARY KEY(customer_id) );
ALTER TABLE customers
  ADD CONSTRAINT customers_customer_id_pk
      PRIMARY KEY(customer_id);
```

- Name of the table
- Actual constraint name
- A suffix that identifies the constraint type

Component value	Description
customers	Table name
_customer_id	Constraint name (usually the column name)
_pk	Constraint type (primary key in this example)

Suffix for Constraint Names

Constraint	Suffix
PRIMARY KEY	_pk
FOREIGN KEY	_fk
UNIQUE	_uq
CHECK	_ck
NOT NULL	_nn

Unique Constraints

- Must be a unique column value
- Can be NULL

Unique Constraint Example

```
CREATE TABLE employee_jobs
  ( employee_id     INTEGER         NOT NULL,
    first_name      CHAR     (15)   NOT NULL,
    middle_initial  CHAR     (1)    NOT NULL,
    last_name       CHAR     (15)   NOT NULL,
    soc_sec_nbr     INTEGER         NOT NULL,
    birth_date      DATE            NOT NULL,
    hire_date       DATE            NOT NULL,
    work_department CHAR     (2)    NOT NULL,
    phone_ext       SMALLINT        NOT NULL,
    job_class       CHAR     (1)    NOT NULL,
    job_level       CHAR     (1)    NOT NULL,
    sex             CHAR     (1)    NOT NULL,
    salary          DECIMAL  (9,2)  NOT NULL,
    bonus           DECIMAL  (9,2)  NOT NULL,
    commission      DECIMAL  (9,2)  NOT NULL,
CONSTRAINT employee_jobs_employee_id_pk
    PRIMARY KEY(employee_id) );

ALTER TABLE employee_jobs
    ADD CONSTRAINT employee_jobs_soc_sec_nbr_uq
        UNIQUE ( soc_sec_nbr );
```

Foreign Key Constraints

- A **foreign key** is one or more columns in the child (dependent) table that contain values that match the primary key of a parent table

```
CREATE TABLE departments
( department_code     CHAR     (2)   NOT NULL,
  department_name     CHAR     (30)  NOT NULL,
  Location            CHAR     (20)  NOT NULL,
  CONSTRAINT departments_department_code_pk
       PRIMARY KEY( department_code) );

CREATE TABLE employee_jobs
( employee_id     INTEGER         NOT NULL,
  first_name      CHAR     (15)   NOT NULL,
  middle_initial  CHAR     (1)    NOT NULL,
  last_name       CHAR     (15)   NOT NULL,
  soc_sec_nbr     INTEGER         NOT NULL,
  birth_date      DATE            NOT NULL,
  hire_date       DATE            NOT NULL,
  work_department CHAR     (2)    NOT NULL,
  phone_ext       SMALLINT        NOT NULL,
  job_class       CHAR     (1)    NOT NULL,
  job_level       CHAR     (1)    NOT NULL,
  sex             CHAR     (1)    NOT NULL,
  salary          DECIMAL  (9,2)  NOT NULL,
  bonus           DECIMAL  (9,2)  NOT NULL,
  commission      DECIMAL  (9,2)  NOT NULL,
  CONSTRAINT employee_jobs_employee_id_pk
    PRIMARY KEY( employee_id) );

ALTER TABLE employee_jobs
  ADD CONSTRAINT employee_jobs_work_department_fk
    FOREIGN KEY ( work_department )
    REFERENCES departments( department_code );
```

Check Constraints

- **Check constraints** are used to enforce business rules by placing restrictions on the data that can be entered into a column

Compare a column to a range of values

```
ALTER TABLE employee_jobs
  ADD CONSTRAINT employee_jobs_salary_range_ck
    CHECK ( salary > 15.00 AND salary < 45.00 );
```

Compare two columns

```
ALTER TABLE employee_jobs
  ADD CONSTRAINT employee_jobs_birth_hire_date_ck
    CHECK ( hire_date > birth_date );
```

List Constraint

```
ALTER TABLE employee_jobs
  ADD CONSTRAINT employee_jobs_job_class_ck
    CHECK ( job_class IN ('T', 'J', 'C', 'M'));
```

```
ALTER TABLE employee_jobs
  ADD CONSTRAINT employee_jobs_job_class_ck
    CHECK (  job_class = 'T'
          OR job_class = 'J'
          OR job_class = 'C'
          OR job_class = 'M');
```

Limit Constraint

```
ALTER TABLE employee_jobs
  ADD CONSTRAINT employee_jobs_salary_ck
    CHECK ( salary < 92000.00 );
```

OR Constraint

```
ALTER TABLE employee_jobs
  ADD CONSTRAINT employee_jobs_comm_bonus_ck
    CHECK ( commission > 0 OR bonus > 0 );
```

NULL Constraint

```
ALTER TABLE orders
  ADD CONSTRAINT orders_ship_date_ck
    CHECK( ship_date IS NULL OR ship_date >= order_date );
```

Compare more than one column Constraint

```
ALTER TABLE shipTBL
  ADD CONSTRAINT shiptbl_status_name_ck
    CHECK ( ( status = 'A' OR status = 'I' )
   AND  ( name  <> ' ' ) );
```

Expressions with Constraints

```
ALTER TABLE employee_jobs
  ADD CONSTRAINT employee_jobs_comm_salary_ck
    CHECK ( commission < salary * .05 ),
```

Boolean Data Type

- DB2 does not support a Boolean data type
- Can be represented using a CHECK constraint

```
status  INTEGER NOT NULL DEFAULT 0,

ALTER TABLE customers
  ADD CONSTRAINT customers_status_ck
    CHECK (status IN (0, 1);
```

Defining Constraints at the Table level

- Not recommended

```
CREATE TABLE employee_jobs
( employee_id     INTEGER        NOT NULL PRIMARY KEY,
  first_name      CHAR    (15)   NOT NULL,
  middle_initial  CHAR    ( 1)   NOT NULL,
  last_name       CHAR    (15)   NOT NULL,
  soc_sec_nbr     INTEGER        NOT NULL,
    CONSTRAINT employee_jobs_soc_sec_nbr_uq
      UNIQUE ( soc_sec_nbr ),
  birth_date    DATE             NOT NULL,
.
.
```

Removing Primary Key Constraint

- Specify the PRIMARY KEY keywords

```
ALTER TABLE employee_jobs
   DROP PRIMARY KEY;
```

Removing other Constraints

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

```
ALTER TABLE employee_jobs
  DROP CONSTRAINT employee_jobs_empsoc_sec_nbr_uq;
```

Adding a Constraint

- When a new table is created with constraints, data does not yet exist in the table
  - Thus all the constraints are enabled immediately

- When the ALTER TABLE command is used to add a constraint after the database table has been populated with data, the command succeeds only if all existing rows satisfy the constraint