Integrity Constraints

Dr. Jenila Livingston L.M.

VIT Chennai

SQL Integrity Constraints

1. **Key Constraints**

- PRIMARY KEY Constraint
  - Ensures that all values in column are unique and NOT NULL
- UNIQUE KEY constraint
  - Ensures that all values in column are unique

2. **Attribute Constraints**

- NOT NULL constraint
  - Ensures that column does not accept nulls
- DEFAULT constraint
  - Assigns value to attribute when a new row is added to table
- CHECK constraint
  - Validates data when attribute value is entered

3. **Referential Integrity Constraints**

- FOREIGN KEY constraint
  - Defines a foreign key for a table

1. Key Constraints

Idea: specifies that a relation is a set, not a bag

SQL examples:

1. **Primary Key:**

```
CREATE TABLE branch(
        bname  CHAR(15)  PRIMARY KEY,
        bcity      CHAR(20),
        assets    INT);
```

2. **Candidate/Unique Keys:**

```
CREATE TABLE customer (
          ssn     CHAR(9)    PRIMARY KEY,
          cname  CHAR(15),
          address CHAR(30),
          city          CHAR(10),
           UNIQUE (cname, address, city);
```

2. Attribute Constraints

Idea:

- Attach constraints to values of attributes
- Enhances types system (e.g.: >= 0 rather than integer)

In SQL:

1. **NOT NULL**

```
e.g.:   CREATE TABLE branch(
               bname   CHAR(15)  NOT NULL,
               ....
               )
```

Note: declaring bname as primary key also prevents null values

2. **CHECK**

```
e.g.:   CREATE TABLE depositor(
                ....
                balance int NOT NULL,
                CHECK( balance >= 0),
                 ....
                 )
```

affect insertions, update in affected columns

Check and default Constraint - Eg

2. Check constraint

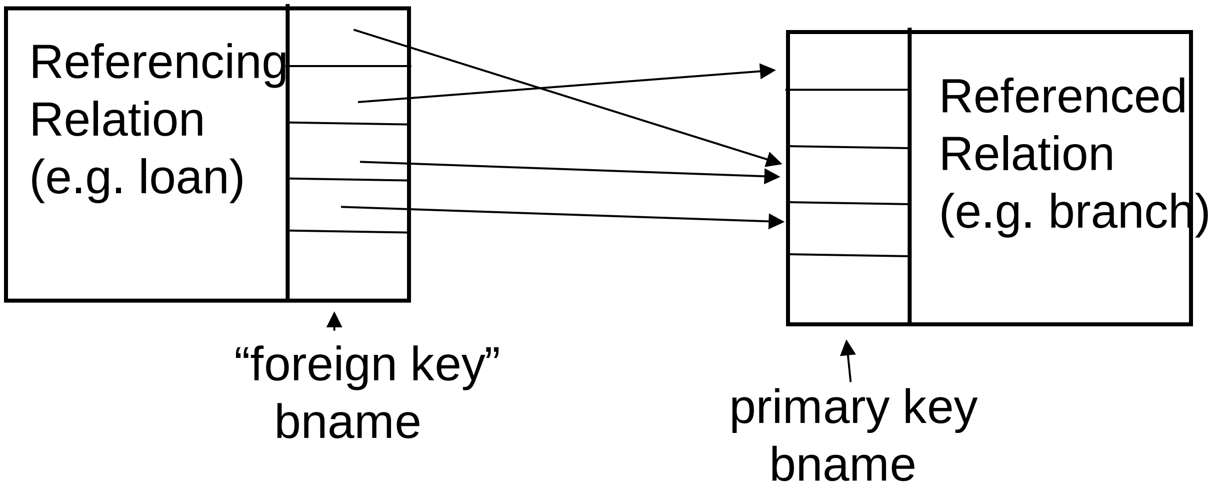
```
CREATE TABLE credit_card(
                   ....
                   balance int NOT NULL,
                    CHECK(  balance >= 0),
                    CHECK (balance < limit),
                    ....
                    )
```

3. Default constraint

```
alter table stu modify gender varchar2(6) default('male');
create table stu1(name varchar2(10),gender varchar2(5)
default('male'));
```

3.Referential Integrity Constraints

Idea: prevent “dangling tuples” (e.g.: a loan with a bname of ‘Kenmore’ when no Kenmore tuple is not in branch table)



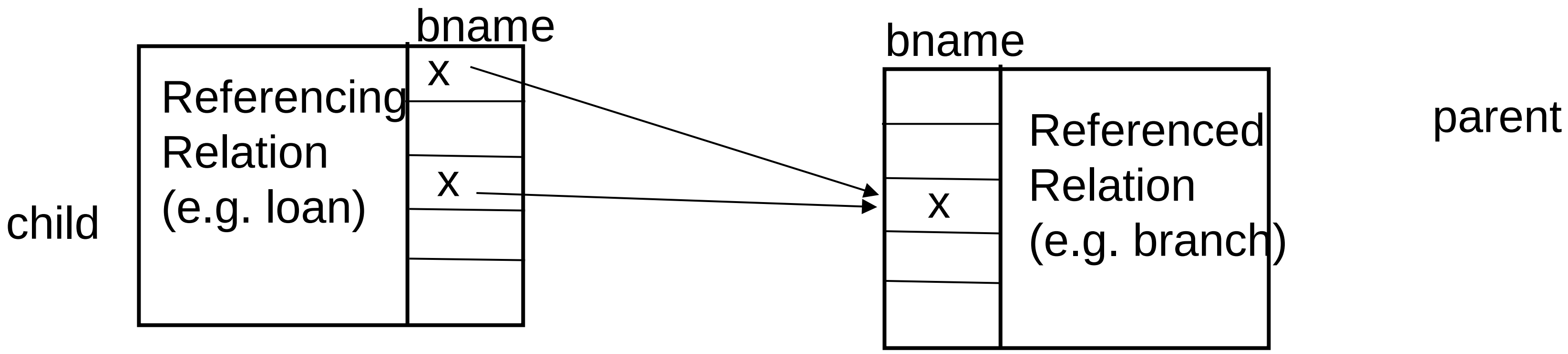
Ref Integrity:

ensure that:

foreign key value → primary key value

(note: need not to ensure ←, i.e., not all branches have to have loans)

Referential Integrity Constraints



bname

x

x

Referencing Relation (e.g. loan)

child

bname

x

Referenced Relation (e.g. branch)

parent

In SQL:

```
CREATE TABLE  branch(
        bname   CHAR(15)   PRIMARY KEY
        ....)
```

```
CREATE TABLE loan (
                    .........
bname REFERENCES branch(bname));
```

- Only primary key can be referenced by “foreign key” (ref integrity)

Affects:

1) Insertions, updates of referencing relation
2) Deletions, updates of referenced relation

Referential integrity

- How referential integrity works
- Cascade Update and Delete

Referential Integrity

- Referential integrity controls the links between records of primary table and reference table

Example

Order Products

OrderID	Product ID	Qty
1	A1	1
1	A2	15
1	A3	23
2	A1	12

Employee

EmployeeID	Employee Name
100	John Smith
101	Mia Lee

Product

ProductID	ProductName
A1	Football
A2	Tennis Ball
A3	Golf Clubs

Customer

CustID	Cust Name
C1	Gianni Albin
C2	Martin Jones

Order table

OrderID	EmployeeID	CustID	OrderDate	Deliver_by	Comments
1	100	C1	29/03/2009	29/04/2009	Ring the bell
2	101	C2	16/04/2009	17/04/2009	N/A

How referential integrity works

- You can't enter a value in the *foreign key* field of a table if it doesn't exist in the primary key of the related table
  - For example, you can't assign an order to a customer who is not in the Customer table
- You can't delete a record from a table if matching records exist in a related table
  - For example, you can't delete an employee if there are orders assigned to the employee in the Order table

How referential integrity works (2)

- You can't update a primary key value for a record that has related records
    - For example, you can't change an employee's ID in the Employee table if there are orders assigned to that employee in the Orders table.

Cascade Update and Delete

- For relationships with referential integrity enforced, you can say whether you want to automatically cascade update and cascade delete related records

- When you try to delete records or update linked primary key values, itchanges the related tables to allow your updates to go ahead, preserving referential integrity

Referential Integrity in SQL

By default, in SQL a foreign key references the primary-key attributes of the referenced table.

dept-name ***references department(dept_name)***

Referential Integrity in SQL

- When a referential-integrity constraint is violated, the normal procedure is to reject the action that caused the violation (that is, the transaction performing the update action is rolled back).

Cascade Update and Delete in SQL

```
create table course
( . . .
foreign key (dept name) references department
on delete cascade
on update cascade,
. . . );
```

On delete not null

Example

Example of cascade update:

if you change a customer's ID in the Customer table, the CustomerID field in the Orders table is automatically updated for that customer's orders so that the relationship isn't broken

Example

Order Products

OrderID	Product ID	Qty
1	A1	1
1	A2	15
1	A3	23
2	A1	12

Employee

EmployeeID	Employee Name
100	John Smith
101	Mia Lee

Product

ProductID	ProductName
A1	Football
A2	Tennis Ball
A3	Golf Clubs

Customer

CustID	Cust Name
C1	Gianni Albin
C2	Martin Jones

Order table

OrderID	EmployeeID	CustID	OrderDate	Deliver_by	Comments
1	100	C1	29/03/2009	29/04/2009	Ring the bell
2	101	C2	16/04/2009	17/04/2009	N/A