CSE 414: Section 7

Siena Dumas Ang

Foreign Key Constraints

- Example with multi-attribute primary key

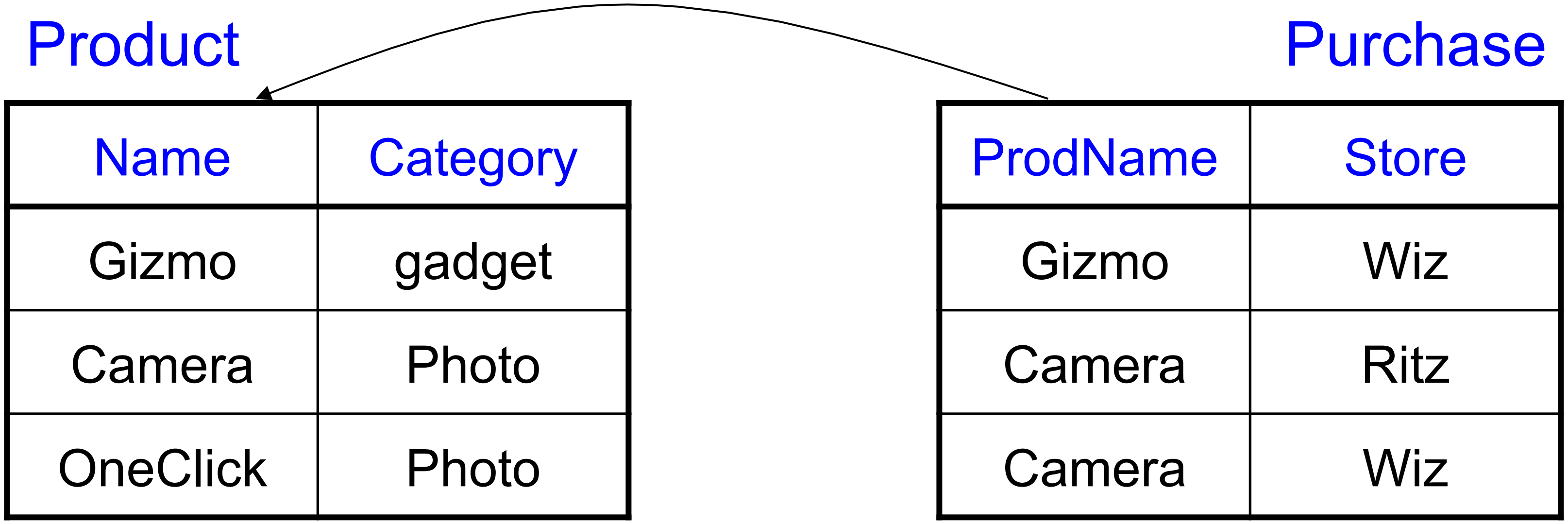
```
CREATE TABLE Purchase (
        prodName CHAR(30),
        category VARCHAR(20),
        date DATETIME,
        FOREIGN KEY (prodName, category)
          REFERENCES  Product(name, category)
```

- (name, category) must be a KEY in Product

What happens when data changes?

Types of updates:

- In Purchase: insert/update
- In Product: delete/update



Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

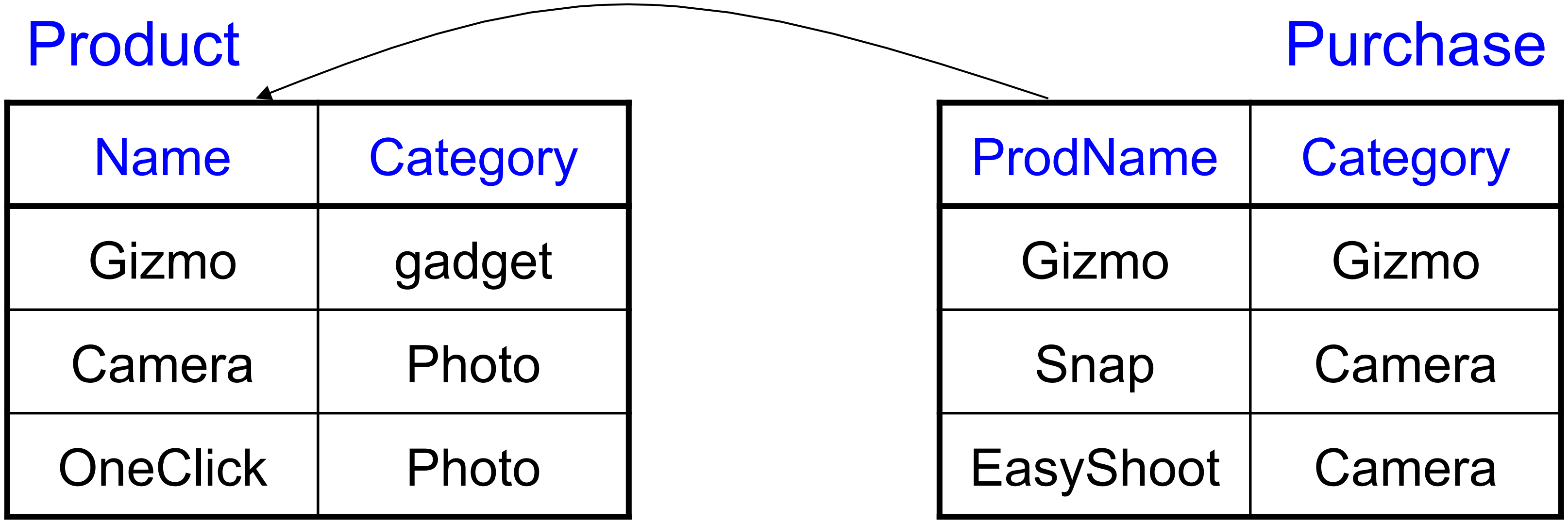
ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

What happens when data changes?

- SQL has three policies for maintaining referential integrity:
- NO ACTION reject violating modifications (default)
- CASCADE after delete/update do delete/update
- SET NULL set foreign-key field to NULL
- SET DEFAULT set foreign-key field to default value
  - need to be declared with column, e.g.,
    CREATE TABLE Product (pid INT DEFAULT 42)

Maintaining Referential Integrity

```
CREATE TABLE Purchase (
        prodName CHAR(30),
        category VARCHAR(20),
        date DATETIME,
        FOREIGN KEY (prodName, category)
          REFERENCES  Product(name, category)
          ON UPDATE CASCADE
          ON DELETE SET NULL   )
```



Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Category
Gizmo	Gizmo
Snap	Camera
EasyShoot	Camera

Constraints on Attributes and Tuples

- Constraints on attributes:

  NOT NULL -- obvious meaning...

  CHECK condition -- any condition !

- Constraints on tuples

  CHECK condition

Constraints on Attributes and Tuples

```
CREATE TABLE R (
            A int NOT NULL,
            B int CHECK (B > 50 and B < 100),
            C varchar(20),
            D int,
            CHECK (C >= 'd' or D > 0))
```

Constraints on Attributes and Tuples

```
CREATE TABLE Product (
        productID  CHAR(10),
        name CHAR(30),
        category VARCHAR(20),
        price INT CHECK (price > 0),
        PRIMARY KEY (productID),
        UNIQUE (name, category))
```

Constraints on Attributes and Tuples

What does this constraint do?

What is the difference from Foreign-Key ?

```
CREATE TABLE Purchase (
        prodName CHAR(30)
                CHECK (prodName IN
                        (SELECT Product.name
                         FROM Product),
        date DATETIME NOT NULL)
```

General Assertions

```
CREATE ASSERTION myAssert CHECK
  (NOT EXISTS(
           SELECT Product.name
           FROM Product, Purchase
           WHERE Product.name = Purchase.prodName
           GROUP BY Product.name
           HAVING count(*) > 200) )
```

But most DBMSs do not implement assertions
Because it is hard to support them efficiently
Instead, they provide triggers

Database Triggers

- Event-Condition-Action rules
- Event
  - Can be insertion, update, or deletion to a relation
- Condition
  - Can be expressed on DB state before or after event
- Action
  - Perform additional DB modifications

More About Triggers

- Row-level trigger
  - Executes once for each modified tuple

- Statement-level trigger
  - Executes once for all tuples that are modified in a SQL statement

Database Triggers Example

When Product.price is updated, if it is decreased then set Product.category = ‘On sale’

```
CREATE TRIGGER ProductCategories
AFTER UPDATE OF price ON Product
REFERENCING
   OLD ROW AS OldTuple
   NEW ROW AS NewTuple
FOR EACH ROW
WHEN (OldTuple.price > NewTuple.price)
        UPDATE Product
        SET category = ‘On sale’
        WHERE productID = OldTuple.productID
```

SQL Server Example

```
CREATE TRIGGER ProductCategory
ON Product
AFTER UPDATE
AS
  BEGIN
    UPDATE Product
    SET category=‘sale’ WHERE productID IN
    (SELECT i.productID from inserted i, deleted d
     WHERE i.productID = d.productID
     AND i.price < d.price)
  END
```

Boyce-Codd Normal Form

There are no "bad" FDs:

Definition. A relation R is in BCNF if:

Whenever $X \rightarrow B$ is a non-trivial dependency, then X is a superkey.

Equivalently:

Definition. A relation R is in BCNF if:

$\forall$ X, either $X^+ = X$ or $X^+ =$ [all attributes]

Problem 1

R(A,B,C,D,E,F,G)

A → D
D → C
F → E,G
D,C → B,F

From A → D, {A}+ = {A,B,C,D,E,F,G}, it is useless
From D→C, {D}+ = {D,C,B,F,E,G}, we can decompose R into R1 = {D,C,B,F,E,G} and R2 = {A,D}
From F → E,G, {F}+ = {F,E,G} so we can further decompose R1 into:

R11 = {E,F,G} and R12 = {C,D,B,F}

Problem 2

R(A,B,C,D,E,F)

A → BC
D → AF

From A → BC, {A}+ = {A,B,C}, since closure is not {A,B,C,D,E,F} this violates BCNF.
So decompose R into R1 = {A,B,C} and R2 = {A,D,E,F}
R1 is in BCNF.
From D→ AF, {D}+ = {D,A,F} which violates BCNF.
So we split R2 into:

R21= {D,A,F} and R22 = {D,E}