PAYAP UNIVERSITY

Advanced SQL

Vaidė Narváez

Computer Information Systems

December 7th, 2010

Queries involving multiple relations

- Interesting queries often combine data from more than one relation.
- We can address several relations in one query by listing them all in the FROM clause.
- Distinguish attributes of the same name by “relation_name.attribute”.

PAYAP UNIVERSITY

Joins in SQL

Find the beers that the frequenters of Joe's Bar like.
Likes(drinker, beer)
Frequents(drinker, bar)

PAYAP UNIVERSITY

Joins in SQL

Find the beers that the frequenters of Joe's Bar like.
Likes(drinker, beer)
Frequents(drinker, bar)

```
SELECT beer
FROM Frequents, Likes
WHERE bar = 'Joe"s Bar' AND
Frequents.drinker = Likes.drinker;
```

PAYAP UNIVERSITY

Self-Joins

Find all pairs of beers by the same manufacturer
(*Do not produce pairs like (Bud, Bud)*)
Beers(name, manufacturer)

```
SELECT b1.name, b2.name
FROM Beers b1, Beers b2
WHERE b1.manufacturer = b2.manufacturer AND b1.name < b2.name;
```

PAYAP UNIVERSITY

SQL Joins

- Join – a relational operation that causes two or more tables with a common domain to be combined into a single table or view
- Equi-join – a join in which the joining condition is based on equality between values in the common columns; common columns appear redundantly in the result table
- Natural join – an equi-join in which one of the duplicate columns is eliminated in the result table
- Outer join – a join in which rows that do not have matching values in common columns are nonetheless included in the result table (as opposed to inner join, in which rows must have matching values in order to appear in the result table)
- Union join – includes all columns from each table in the join, and an instance for each row of each table

PAYAP UNIVERSITY

Joins syntax

```
SELECT beer
FROM Frequents, Likes
WHERE Frequents.drinker = Likes.drinker;
```

or

```
SELECT beer
FROM Frequents INNER JOIN Likes ON
Frequents.drinker = Likes.drinker;
```

PAYAP UNIVERSITY

Subqueries

Result of a SELECT-FROM-WHERE query can be used in the WHERE-clause of another query

Find bars that serve Miller at the same price Joe charges for Bud.
Sells(bar,beer,price)

```
SELECT bar
FROM Sells
WHERE beer = 'Miller' AND price =
(SELECT price
FROM Sells
WHERE bar = 'Joe"s Bar' AND beer = 'Bud');
```

PAYAP UNIVERSITY

IN and NOT IN operators

Find the name and manufacturer of beers that Fred likes.
Beers(name, manufacturer)
Likes(drinker, beer)

```
SELECT *
FROM Beers
WHERE name IN
(SELECT beer
FROM Likes
WHERE drinker = 'Fred);
```

PAYAP UNIVERSITY

EXISTS and NOT EXISTS

EXISTS(subquery) is true if and only if the subquery result is not empty.
Find the beers that are the unique beer by their manufacturer
Beers(name, manufacturer)

```
SELECT name
FROM Beers b1
WHERE NOT EXISTS
(SELECT *
FROM Beers
WHERE manufacturer = b1.manufacturer AND name <> b1.name);
```

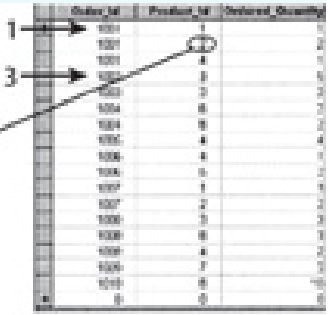
PAYAP UNIVERSITY

Subqueries

- correlated subqueries – executed once for each row returned by the outer query
- non-correlated subqueries – executed once for the entire outer query

Correlated subqueries

```
SELECT DISTINCT ORDER _ID FROM ORDER _LINE _T
WHERE EXISTS
        (SELECT *
          FROM PRODUCT _T
                WHERE PRODUCT _ID = ORDER _LINE _T.PRODUCT _ID
                    AND PRODUCT _FINISH = 'Natural Ash');
```



	Product_ID	Product_Description	Product_Finish	Standard_Price	Product_Line_Id
	1	End Table	Cherry	$175.00	10001
2 →	2	Coffee Table	Natural Ash	$200.00	20001
4 →	3	Computer Desk	Natural Ash	$375.00	20001
	4	Entertainment Center	Natural Maple	$650.00	30001
	5	Writer's Desk	Cherry	$325.00	10001
	6	8-Drawer Dresser	White Ash	$750.00	20001
	7	Dining Table	Natural Ash	$800.00	20001
	8	Computer Desk	Walnut	$250.00	30001
	(AutoNumber)			$0.00	

1. The first order ID is selected from ORDER _LINE _T: ORDER _ID =1001.
2. The subquery is evaluated to see if any product in that order has a natural ash finish. Product 2 does, and is part of the order. EXISTS is valued as *true* and the order ID is added to the result table.
3. The next order ID is selected from ORDER _LINE _T: ORDER _ID =1002.
4. The subquery is evaluated to see if the product ordered has a natural ash finish. It does. EXISTS is valued as *true* and the order ID is added to the result table.
5. Processing continues through each order ID. Orders 1004, 1005, and 1010 are not included in the result table because they do not include any furniture with a natural ash finish. The final result table is shown in the text on page 334.

PAYAP UNIVERSITY

ANY, ALL quantifiers

Find the beer(s) most expensive beer.
Sells(bar,beer,price)

```
SELECT beer
FROM Sells
WHERE price >= ALL(
SELECT price FROM Sells);
```