CS 186 Discussion #2

SQL Queries

Logistics

- HW1 is out…
  - Due Friday 11:59 pm

Revisiting External Hashing…

- Vitamin #1 Q4
  - Suppose something went wrong in Q3's hashing and one of the partitions is 70 MB (greater than the 64 MB allocated buffer). What's the additional I/O cost of having to externally hash this single partition?
    - Assume that a new hashing function is chosen for the second pass such that the records are distributed in a way that guarantees subsequent partitions to be under 64 MB. (Hint: be aware of when this partition is conquered)

SQL Queries

SELECT [DISTINCT] <column list>
FROM <table1>
WHERE <predicate>
GROUP BY <column list>
HAVING <predicate>
ORDER BY <column list> [DESC/ASC]
LIMIT <amount>

Also Review…

- Nested Queries
  - VIEWs, WITH
- UNION/INTERSECT
- Set Comparison Operators
  - IN, EXISTS, ANY, ALL
- Primary Keys
  - And Foreign Keys, Candidate Keys, etc…

Live Demo

Worksheet 1, 2, 3, 4

1. Five songs that spent the most time in the top 40:

Worksheet 1, 2, 3, 4

1. Five songs that spent the most time in the top 40:

```
SELECT song_name
FROM   Songs
ORDER BY weeks_in_top_40 DESC
LIMIT 5;
```

Worksheet 1, 2, 3, 4

2. Name and first year active of every artist whose name starts with ‘B’:

Worksheet 1, 2, 3, 4

2. Name and first year active of every artist whose name starts with ‘B’:

```
SELECT artist_name, first_year_active
FROM   Artists
WHERE artist_name LIKE ‘B%’;
```

Worksheet 1, 2, 3, 4

3. Total number of ‘Techno’ albums released each year:

Worksheet 1, 2, 3, 4

3. Total number of ‘Techno’ albums released each year:

```
SELECT    year_released, COUNT(*)
FROM      Albums
WHERE     genre = ‘Techno’
GROUP BY year_released;
```

Worksheet 1, 2, 3, 4

4. Number of albums per genre, ignoring genres with fewer than 10 albums:

Worksheet 1, 2, 3, 4

4. Number of albums per genre, ignoring genres with fewer than 10 albums:

```
SELECT     genre, COUNT(*)
FROM       Albums
GROUP BY year_released
HAVING     COUNT(*) >= 10;
```

SQL Queries

SELECT [DISTINCT] <column list>
FROM <table1> AS A, <table2> AS B
WHERE <predicate>
GROUP BY <column list>
HAVING <predicate>
ORDER BY <column list>
LIMIT <amount>

Worksheet 5, 6, 7, 8

5. All ‘Country’ songs that have been in the top 40 for more than 2 weeks:

Worksheet 5, 6, 7, 8

5. All ‘Country’ songs that have been in the top 40 for more than 2 weeks:

```
SELECT song_name
FROM   Songs S, Albums A
WHERE S.album_id = A.album_id
    AND  genre = ‘Country’
    AND  weeks_in_top_40 > 14;
```

What kind of join is this?

Worksheet 5, 6, 7, 8

6. For each song… song name, album name, and artist name:

Worksheet 5, 6, 7, 8

6. For each song… song name, album name, and artist name:

```
SELECT song_name, album_name, artist_name
FROM   Songs S, Albums A, Artists R
WHERE S.album_id = A.album_id
   AND  A.artist_id = R.artist_id;
```

Worksheet 5, 6, 7, 8

7. Number of albums from each artist:

Worksheet 5, 6, 7, 8

7. Number of albums from each artist:

```
SELECT    artist_name, COUNT(*)
FROM      Albums A, Artists R
WHERE     A.artist_id = R.artist_id
GROUP BY R.artist_id, artist_name;
```

Why do we need to group by artist_id?

Worksheet 5, 6, 7, 8

8. Singers with both ‘Pop’ and ‘Techno’:

Worksheet 5, 6, 7, 8

8. Singers with both ‘Pop’ and ‘Techno’:

```
SELECT    artist_name
FROM      Albums A1, Albums A2, Artists R
WHERE     R.artist_id = A1.artist_id
  AND     A1.artist_id = A2.artist_id
  AND     A1.genre = ‘Pop’
  AND     A2.genre = ‘Techno’;
```

SQL Joins

Inner/Natural Join

TABLE Students

SID	Name
23357852	Bob the Builder
24821529	Anonymous
27983421	Nicholas
27994495	Oskicat

TABLE Enrollment

CCN	SID
25227	23357852
26586	23357852
21545	21592421
23495	23374219
26586	27994495

SID	Name	CCN	*SID*
23357852	Bob the Builder	25227	*23357852*
23357852	Bob the Builder	26586	*23357852*
27994495	Oskicat	26586	*27994495*

Inner/Natural Join

```
SELECT <column list>
FROM <table1> A, <table2> B
WHERE A.<column> = B.<column>
```

```
SELECT <column list>
FROM <table1> A INNER JOIN <table2> B
ON A.<column> = B.<column>
```

```
SELECT <column list>
FROM <table1> A NATURAL JOIN <table2> B
```

Outer Joins

SELECT <column list>

FROM A {LEFT, RIGHT, FULL} OUTER JOIN B

ON A.<column> = B.<column>

Left Outer Join

TABLE Students

SID	Name
23357852	Bob the Builder
24821529	Anonymous
27983421	Nicholas
27994495	Oskicat

TABLE Enrollment

CCN	SID
25227	23357852
26586	23357852
21545	21592421
23495	23374219
26586	27994495

SID	Name	CCN	SID
23357852	Bob the Builder	25227	23357852
23357852	Bob the Builder	26586	23357852
24821529	Anonymous		
27983421	Nicholas		
27994495	Oskicat	26586	27994495

Right Outer Join

TABLE Students

SID	Name
23357852	Bob the Builder
24821529	Anonymous
27983421	Nicholas
27994495	Oskicat

TABLE Enrollment

CCN	SID
25227	23357852
26586	23357852
21545	21592421
23495	23374219
26586	27994495

SID	Name	CCN	SID
23357852	Bob the Builder	25227	23357852
23357852	Bob the Builder	26586	23357852
		21545	21592421
		23495	23374219
27994495	Oskicat	26586	27994495

Full Outer Join

SID	Name	CCN	SID
23357852	Bob the Builder	25227	23357852
23357852	Bob the Builder	26586	23357852
24821529	Anonymous		
27983421	Nicholas		
		21545	21592421
		23495	23374219
27994495	Oskicat	26586	27994495