OLAP

Harman Gill
SJSU

Content from Database Systems-The Complete Book Dr. Jennifer Widom and Oracle SQL guide

OLTP vs OLAP

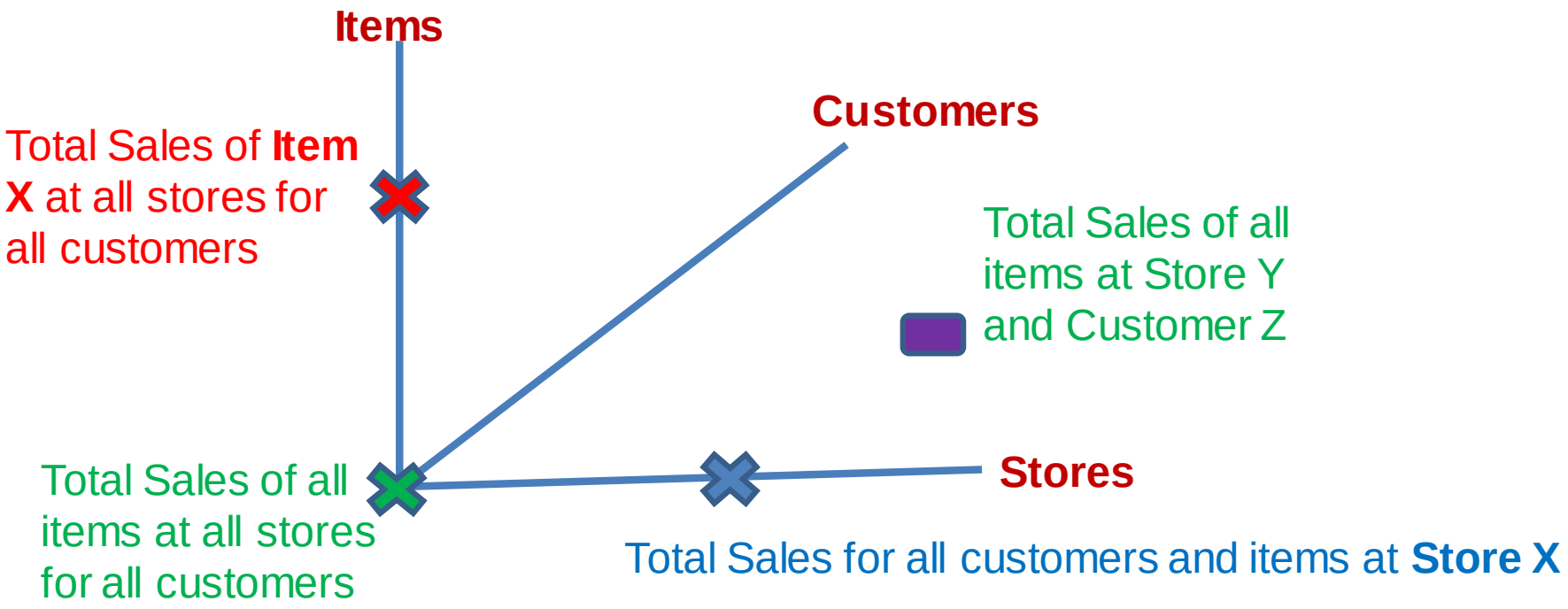
- Online Transaction Processing
  - Short transaction and simple queries
  - Queries/Updates touch small portion of data
  - Updates are frequent
- Online Analytical Processing
  - Long transaction and complex queries
  - Queries touch large portion of the data
  - Updates are infrequent

Database Warehousing

- Software architecture that brings the data from distributed OLTP sources into a single gigantic warehouse
  - Warehouse is used for analysis of the data
    - All statistics about all sales
      - Group by department and month
    - Example: Retail Chain Store copies all distributor, store, sales information into data warehouse. Analysis on terabytes of data to make supplier decisions or store layout.

Data Cube

- Multidimensional OLAP
  - Dimensions attributes form axes of cube
  - Aggregated data on sides, edges, and corner
  - Cell has fact data



Use Case

```
SELECT department_id, job_id, SUM(salary), COUNT(employee_id)
FROM employee
GROUP BY department_id, job_id;
```

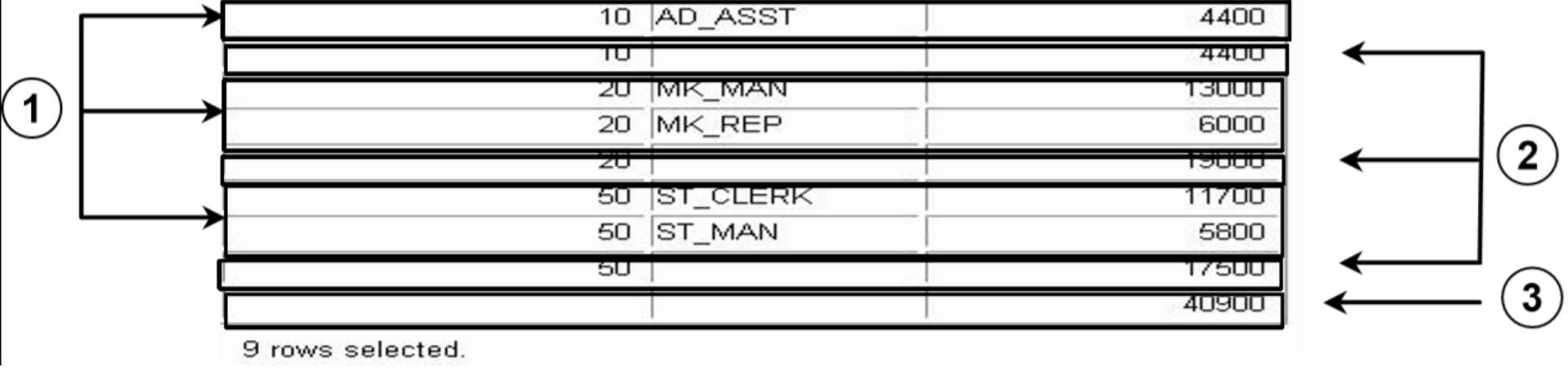
DEPARTMENT_ID	JOB_ID	SUM(SALARY)	COUNT(EMPLOYEE_ID)
10	AD_ASST	4400	1
20	MK_MAN	13000	1
20	MK_REP	6000	1
50	ST_CLERK	11700	4
90	AD_VP	34000	2
110	AC_ACCOUNT	8300	1
110	AC_MGR	12000	1
	SA_REP	7000	1

Use Case – WITH ROLLUP

- Extension to the GROUP BY clause
- Produces cumulative aggregates such as subtotals
- GROUP BY ROLLUP(column1, column2)
  - n = (2), n is number of expressions in the ROLLUP operator of the GROUP BY clause
  - Results in n+1 (3) groupings
    - 1 group with regular GROUP BY rows
    - n groups with superaggregate rows

Use Case – WITH ROLLUP…

```
SELECT   department_id, job_id, SUM(salary)
FROM     employees
WHERE department_id < 60
GROUP BY  ROLLUP(department_id, job_id);
```



DEPARTMENT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
10		4400
20	MK_MAN	13000
20	MK_REP	6000
20		19000
50	ST_CLERK	11700
50	ST_MAN	5800
50		17500
		40900

9 rows selected.

Labeled 1: Regular GROUP BY rows

Labeled 2 and 3: ROLLUP rows

(2) Total salary for those departments whose department ID is less than 60

(3) Total salary for those departments whose department ID is less than 60, irrespective of the job IDs

Use Case – WITH CUBE

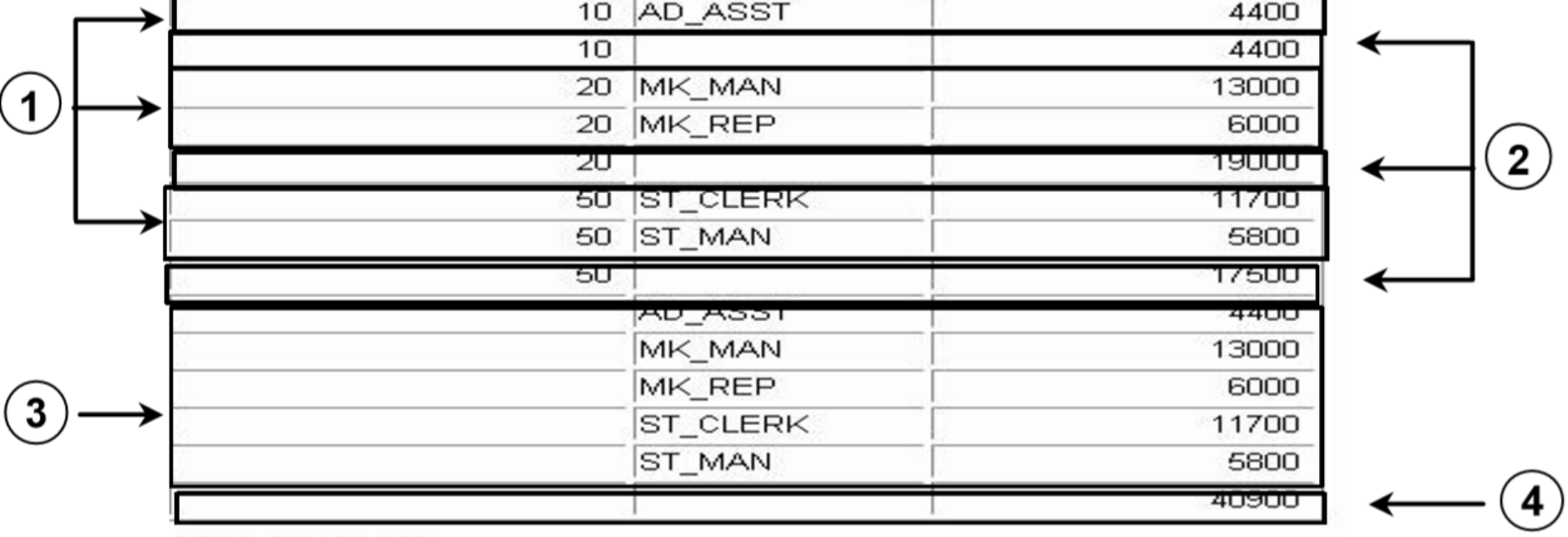
- Extension to the GROUP BY clause
- Produces cross tabulation values
- GROUP BY CUBE(column1, column2)
  - n = 2, n is number of expressions in the CUBE operator of the GROUP BY clause
  - Results in 2^n groupings

Use Case – WITH CUBE…

```
SELECT  department_id, job_id, SUM(salary)
FROM    employees
WHERE   department_id < 60
GROUP BY CUBE (department_id, job_id);
```

DEPARTMENT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
10		4400
20	MK_MAN	13000
20	MK_REP	6000
20		19000
50	ST_CLERK	11700
50	ST_MAN	5800
50		17500
	AD_ASST	4400
	MK_MAN	13000
	MK_REP	6000
	ST_CLERK	11700
	ST_MAN	5800
		40900

14 rows selected.



Labeling 3: Total Salary of every job irrespective of the department