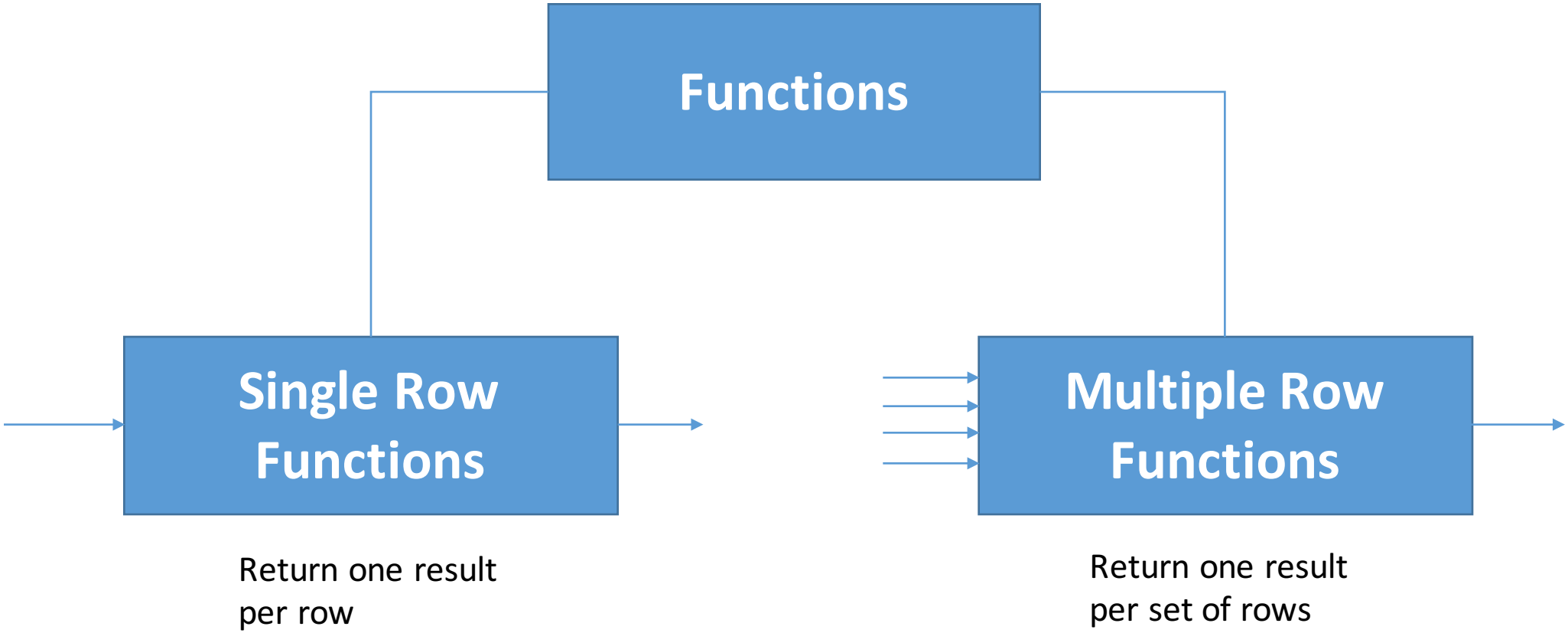
Session-3
Using single-row/multiple row functions to customize output

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

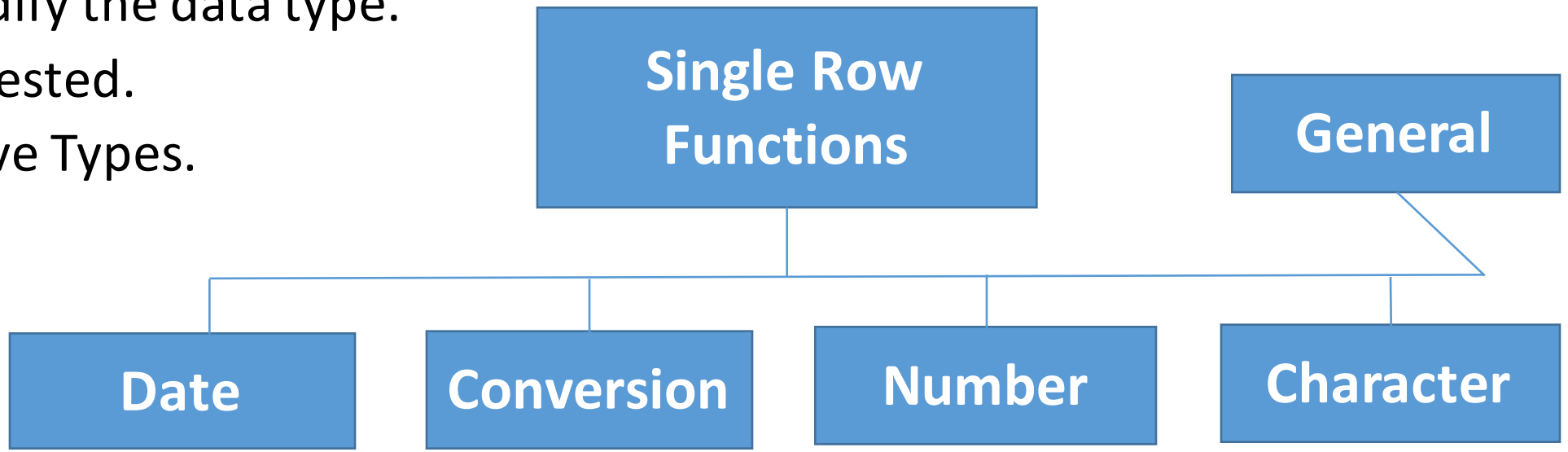
Types of SQL Functions



Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Single Row Functions

- Manipulate data items.
- Accepts arguments and return one value.
- Act on each row that is returned.
- Return one result per row.
- May modify the data type.
- Can be nested.
- Are of Five Types.



Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Single Row Functions

General	Date	Conversion	Character	Number
NVL	ADD_MONTHS	TO_DATE	LOWER	ROUND(number)
NVL2	CURRENT_DATE	TO_NUMBER	UPPER	TRUNC(number)
NULLIF	MONTHS_BETWEEN	TO_CHAR(character)	INITCAP	MOD
COALESCE	NEXT_DAY	TO_CHAR(number)	CONCAT	ABS
CASE	SYSDATE	TO_CHAR(datetime)	SUBSTR	CEIL
DECODE	SYSTIMESTAMP		LENGTH	FLOOR
UID	TO_CHAR		INSTR	
USER	TRUNC(date)		LPAD	
	ROUND(date)		RPAD	
	NUMTOYMINTERVAL		TRIM	
	REMAINDER		REPLACE	
			SOUNDEX	

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Number Functions

```
SELECT ABS(-15) "Absolute" FROM DUAL;          ->15

SELECT order_total, CEIL(order_total)
FROM orders WHERE order_id = 2434;

SELECT FLOOR(15.7) "Floor" FROM DUAL;          ->15

SELECT MOD(11,4) "Modulus" FROM DUAL;          ->3

SELECT ROUND(15.193,1) "Round" FROM DUAL;      ->15.2

SELECT ROUND(15.193,-1) "Round" FROM DUAL;     ->20
```


Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Character Functions

- SELECT LOWER('MR. SCOTT MCMILLAN') "Lowercase" FROM DUAL;
- SELECT INITCAP('the soap') "Capitals" FROM DUAL;
- SELECT LPAD('Page 1',15,'*.') "LPAD example" FROM DUAL;
- SELECT product_name, LTRIM(product_name, 'Monitor ') "Short Name" FROM products WHERE product_name LIKE 'Monitor%';
- SELECT REPLACE('JACK and JUE','J','BL') "Changes" FROM DUAL;
- SELECT last_name, RPAD(' ', salary/1000/1, '*') "Salary" FROM employees WHERE department_id = 80 ORDER BY last_name;

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Character Functions

- SELECT last_name, first_name FROM hr.employees WHERE SOUNDEX(last_name) = SOUNDEX('SMYTHE');

- SELECT SUBSTR('ABCDEFG',3,4) "Substring" FROM DUAL;

- SELECT SUBSTR('ABCDEFG',-5,4) "Substring" FROM DUAL;

- SELECT employee_id, TO_CHAR(TRIM(LEADING 0 FROM hire_date)) FROM employees WHERE department_id = 60;

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Date Functions

- SELECT TO_CHAR (SYSDATE, 'MM-DD-YYYY HH24:MI:SS') "NOW" FROM DUAL;
- SELECT TRUNC(TO_DATE('27-OCT-92','DD-MON-YY'), 'YEAR') "New Year" FROM DUAL;
- SELECT ROUND (TO_DATE ('27-OCT-00'),'YEAR') "New Year" FROM DUAL;
- SELECT NEXT_DAY('02-FEB-2001','TUESDAY') "NEXT DAY" FROM DUAL;

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Date Functions

- SELECT last_name, hire_date, TO_CHAR(ADD_MONTHS(LAST_DAY(hire_date), 5)) "Eval Date" FROM employees;

- SELECT TO_CHAR(ADD_MONTHS(hire_date,1), 'DD-MON-YYYY') "Next month" FROM employees WHERE last_name = 'Baer';

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Conversion Functions

- SELECT TO_CHAR('01110' + 1) FROM dual;

- SELECT TO_DATE('January 15, 1989, 11:00 A.M.', 'Month dd, YYYY, HH:MI A.M.', 'NLS_DATE_LANGUAGE = American') FROM DUAL;

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

General Functions

- SELECT e.last_name, NULLIF(e.job_id, j.job_id) "Old Job ID" FROM employees e, job_history j WHERE e.employee_id = j.employee_id ORDER BY last_name;

- SELECT last_name, NVL(TO_CHAR(commission_pct), 'Not Applicable') "COMMISSION" FROM employees WHERE last_name LIKE 'B%' ORDER BY last_name;

- SELECT last_name, salary, NVL2(commission_pct, salary + (salary * commission_pct), salary) income FROM employees WHERE last_name like 'B%' ORDER BY last_name;

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

General Functions

- SELECT UID FROM DUAL;
- SELECT USER, UID FROM DUAL;
- SELECT USERENV('LANGUAGE') "Language" FROM DUAL;

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

General Functions

- DECODE compares expr to each search value one by one. If expr is equal to a search, then Oracle Database returns the corresponding result. If no match is found, then Oracle returns default. If default is omitted, then Oracle returns null.

```
SELECT product_id,
DECODE (warehouse_id,  1, 'Southlake',
                          2, 'San Francisco',
                          3, 'New Jersey',
                          4, 'Seattle',
                          'Non domestic') "Location of inventory"
FROM inventories
WHERE product_id < 1775;
```


Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

General Functions

- CASE expressions let you use IF ... THEN ... ELSE logic in SQL statements without having to invoke procedures.

```
SELECT  last_name,
CASE department_id
    WHEN 90 THEN 'QA'
    WHEN 100 THEN 'Billing'
    WHEN 30 THEN 'Finance'
    WHEN 50 THEN 'Accounts'
    WHEN 80 THEN 'Marketing'
    ELSE 'No Department'
    END
FROM employees;
```


Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

General Functions

- COALESCE returns the first non-null ***expr*** in the expression list. You must specify at least two expressions. If all occurrences of ***expr*** evaluate to null, then the function returns null.

```
SELECT product_id, list_price, min_price,
COALESCE(0.9*list_price, min_price, 5) "Sale“
 FROM product_information
WHERE supplier_id = 102050
ORDER BY product_id, list_price, min_price, "Sale";
```


Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Multiple Row/Aggregate Functions

- Group functions/aggregate functions operates on sets of rows to give one result per group.

- All group functions ignore null values.

- Functions Are:

AVG, MIN, MAX, COUNT, SUM, STDDEV, MEDIAN, VARIANCE and so on.

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Aggregate Functions

- SELECT AVG(salary) "Average" FROM employees;
- SELECT COUNT(DISTINCT manager_id) "Managers" FROM employees;
- SELECT COUNT(*) "Total" FROM employees;
- SELECT manager_id, last_name, salary, MAX(salary) OVER (PARTITION BY manager_id) AS mgr_max FROM employees;

SELECT manager_id, employee_id, salary, MEDIAN(salary) OVER (PARTITION BY manager_id) "Median by Mgr" FROM employees WHERE department_id > 60;

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Aggregate Functions

- SELECT MIN(hire_date) "Earliest" FROM employees;
- SELECT SUM(salary) "Total" FROM employees group by manager_id;
- SELECT SUM(salary) "Total" FROM employees group by manager_id;
- SELECT last_name, salary, STDDEV(salary) OVER (ORDER BY hire_date) "StdDev" FROM employees WHERE department_id = 30;
- SELECT department_id, job_id, SUM(salary)

  FROM employees GROUP BY department_id, job_id;

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Illegal Queries Using Group Functions

- You cannot use WHERE clause to restrict groups.
- You use the HAVING clause to restrict groups.
- You cannot use group functions in the WHERE clause.
- Below query will through error:

```
SELECT department_id, AVG(salary)
FROM employees
WHERE avg(salary) > 8000
GROUP BY department_id;
```


Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Restricting Group results With HAVING Clause

- When using HAVING clause

1. Rows are grouped.
2. The group function is applied.
3. Groups matching the HAVING clause are displayed.

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id
HAVING avg(salary) > 8000
ORDER BY avg(salary) DESC;
```


Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Nesting Group Functions

```
SELECT MAX(AVG(salary))
FROM employees
GROUP BY department_id
HAVING avg(salary) > 8000;
```


Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Session-3
END

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com