



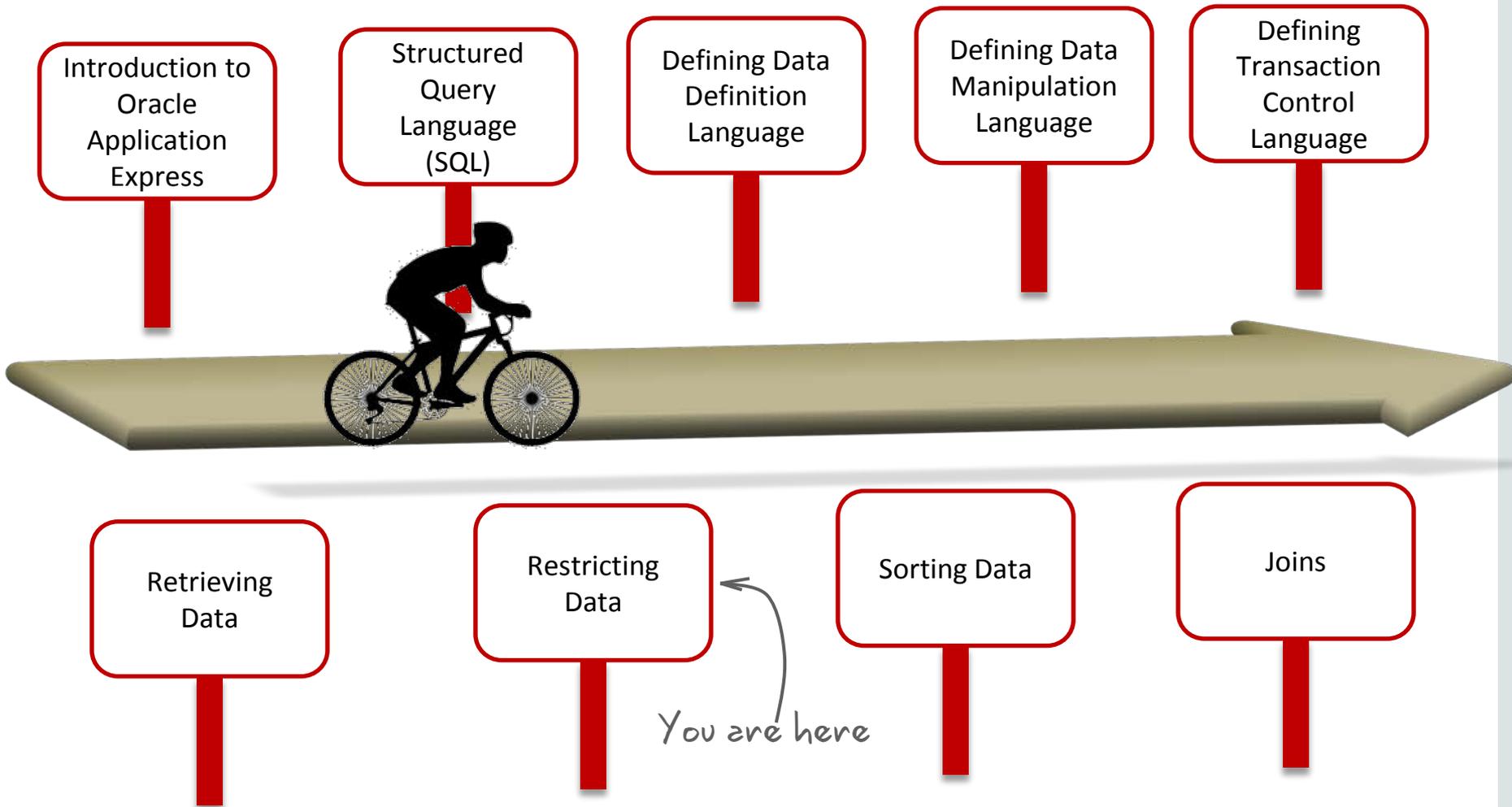
Database Foundations

6-7

Restricting Data Using WHERE



Roadmap



Objectives

This lesson covers the following objectives:

- Limit rows with:
 - WHERE clause
 - Comparison operators using = , <= , >= , <> , > , < , != , ^= , BETWEEN , IN , LIKE , and NULL conditions
 - Logical conditions using AND , OR , and NOT operators
- Describe the rules of precedence for operators in an expression



Limiting Rows Using a Selection

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	SALARY	DEPARTMENT_ID
100	Steven	King	AD_PRES	24000	90
101	Neena	Kochhar	AD_VP	17000	90
102	Lex	De Haan	AD_VP	17000	90
103	Alexander	Hunold	IT_PROG	9000	60
104	Bruce	Ernst	IT_PROG	6000	60
105	David	Austin	IT_PROG	4800	60
106	Valli	Pataballa	IT_PROG	4800	60
107	Diana	Lorentz	IT_PROG	4200	60
108	Nancy	Greenberg	FI_MGR	12008	100

...

"retrieve all employees
in department 90"



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	SALARY	DEPARTMENT_ID
100	Steven	King	AD_PRES	24000	90
101	Neena	Kochhar	AD_VP	17000	90
102	Lex	De Haan	AD_VP	17000	90

Limiting the Rows That Are Selected

- Restrict the rows that are returned by using the `WHERE` clause:

```
SELECT * | { [DISTINCT] column/expression [alias], ... }  
FROM table  
[WHERE logical expression(s)];
```

- The `WHERE` clause follows the `FROM` clause.

Using the WHERE Clause

```
SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, DEPARTMENT_ID
FROM   employees
WHERE  DEPARTMENT_ID = 90 ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

Character Strings and Dates

- Character strings and date values are enclosed in single quotation marks.
- Character values are case-sensitive, and date values are format-sensitive.
- The default date display format is DD-MON-RR .

```
SELECT last_name, job_id, department_id
FROM employees
WHERE last_name = 'Whalen' ;
```

```
SELECT last_name
FROM employees
WHERE hire_date = '17-OCT-03' ;
```

Comparison Operators

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to
BETWEEN...AND...	Between two values (inclusive)
IN (set)	Match any of a list of values
LIKE	Match a character pattern
IS NULL	Is a null value

Using Comparison Operators

```
SELECT last_name, salary
FROM employees
WHERE salary <= 3000 ;
```

LAST_NAME	SALARY
Baida	2900
Tobias	2800
Himuro	2600
Colmenares	2500
Mikkilineni	2700
Landry	2400
Markle	2200

Range Conditions: BETWEEN Operator

Use the BETWEEN operator to display rows based on a range of values:

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500 ;
```

Lower limit

Upper limit

LAST_NAME	SALARY
Khoo	3100
Baida	2900
Tobias	2800
Himuro	2600
Colmenares	2500
Nayer	3200

Case Scenario: Retrieving Data

How do I find book transactions that took place in June?



Faculty

```
SELECT * FROM BOOK_TRANSACTION
WHERE TRANSACTION_DATE
BETWEEN '01-JUN-14' AND '30-JUN-14';
```

TRANSACTION_ID	TRANSACTION_DATE	TRANSACTION_TYPE	BOOK_ID	MEMBER_ID
0D0001	12-JUN-14		BN0012	CN0001
0D0002	28-JUN-14		BN0009	CN0002

Successful retrieval of data



Student

Membership Conditions: IN Operator

Use the IN operator to test for values in a list:

```
SELECT employee_id, last_name, salary, manager_id
FROM employees
WHERE manager_id IN (100, 101, 201) ;
```

EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
101	Kochhar	17000	100
102	De Haan	17000	100
114	Raphaely	11000	100
120	Weiss	8000	100
121	Fripp	8200	100
122	Kaufling	7900	100

Membership Conditions: NOT IN Operator

Use the NOT IN operator to test for values not in a list:

```
SELECT employee_id, last_name, salary, manager_id
FROM employees
WHERE department_id NOT IN (60, 90, 100) ;
```

EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID
114	Raphaely	11000	30
115	Khoo	3100	30
116	Baida	2900	30
117	Tobias	2800	30
118	Himuro	2600	30
119	Colmenares	2500	30
120	Weiss	8000	50
121	Fripp	8200	50
122	Kaufling	7900	50
123	Vollman	6500	50

Pattern Matching: LIKE Operator

- Use the `LIKE` operator to perform wildcard searches of valid search string values.
- Search conditions can contain literal characters or numbers:
 - `%` denotes zero or more characters.
 - `_` denotes one character.

```
SELECT first_name
FROM employees
WHERE first_name LIKE 'S%';
```

Combining Wildcard Characters

- You can combine the two wildcard characters (`%`, `_`) with literal characters for pattern matching:

```
SELECT last_name  
FROM employees  
WHERE last_name LIKE '_o%' ;
```

LAST_NAME
Colmenares
Doran
Fox
Johnson

- You can use the `ESCAPE` identifier to search for the actual `%` and `_` symbols.

Using the NULL Conditions

Test for nulls with the `IS NULL` operator:

```
SELECT last_name, manager_id
FROM employees
WHERE manager_id IS NULL ;
```

LAST_NAME	MANAGER_ID
King	-

Defining Conditions Using the Logical Operators

Operator	Meaning
AND	Returns TRUE if both component conditions are TRUE
OR	Returns TRUE if either component condition is TRUE
NOT	Returns TRUE if the condition is FALSE Returns FALSE if the condition is TRUE

Using the AND Operator

AND requires both component conditions to be true:

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >= 10000
AND job_id LIKE '%MAN%' ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
114	Raphaely	PU_MAN	11000
145	Russell	SA_MAN	14000
146	Partners	SA_MAN	13500
147	Errazuriz	SA_MAN	12000
148	Cambrault	SA_MAN	11000
149	Zlotkey	SA_MAN	10500
201	Hartstein	MK_MAN	13000

Using the OR Operator

OR requires either component condition to be true:

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary >= 10000
OR     job_id LIKE '%MAN%' ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
108	Greenberg	FI_MGR	12008
114	Raphaely	PU_MAN	11000

Using the NOT Operator

```
SELECT last_name, job_id
FROM employees
WHERE job_id
      NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP') ;
```

LAST_NAME	JOB_ID
Baer	PR_REP
Baida	PU_CLERK
Bell	SH_CLERK
Bull	SH_CLERK
Cabrio	SH_CLERK
Cambrault	SA_MAN
Chen	FI_ACCOUNT
Chung	SH_CLERK

Case Scenario: Retrieving Data Using Logical Operators

How do I find the books published by "Elsevier" and "Penguin Group"?

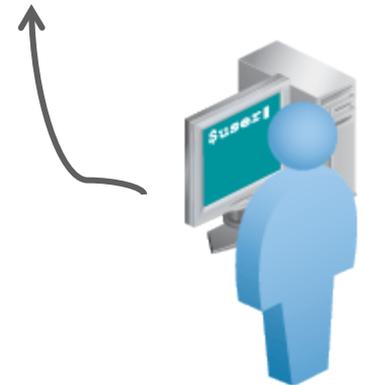


Faculty

```
SELECT * FROM BOOKS
WHERE PUBLISHER_ID = 'PN0001' OR PUBLISHER_ID = 'PN0003';
```

BOOK_ID	TITLE
BN0002	A Vision
BN0003	Citizen of the World
BN0004	The Complete Poetical Works of Oliver Goldsmith
BN0010	Bride of Frankenstein
BN0011	Shelley Poetry and Prose

Successful retrieval
of the book details



Student

Rules of Precedence

Precedence	Operator
1	Arithmetic operators
2	Concatenation operator
3	Comparison conditions
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Not equal to
7	NOT logical operator
8	AND logical operator
9	OR logical operator

Use parentheses to override rules of precedence.

Rules of Precedence

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id = 'SA_REP'
OR job_id = 'AD_PRES'
AND salary > 15000;
```

Precedence of the
AND Operator

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Tucker	SA_REP	10000
Bernstein	SA_REP	9500

```
SELECT last_name, job_id, salary
FROM employees
WHERE (job_id = 'SA_REP'
OR job_id = 'AD_PRES')
AND salary > 15000;
```

Parentheses

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000

Summary

In this lesson, you should have learned how to:

- Limit rows with:
 - WHERE clause
 - Comparison operators using = , <= , >= , <> , > , < , != , ^= , BETWEEN , IN , LIKE , and NULL conditions
 - Logical conditions using AND , OR , and NOT operators
- Describe the rules of precedence for operators in an expression



