



# **Ограничение и сортировка данных**

# Цели

Изучив материалы этого занятия, вы освоите следующие темы:



- Ограничение числа строк, извлекаемых по запросу
- Сортировка строк, извлекаемых по запросу
- Использование переменных подстановки с амперсандами для ограничения и сортировки вывода во время выполнения

# План занятия

- Средства ограничения числа строк:
  - Предложение `WHERE`
  - Условия сравнения с операторами `=`, `<=`, `BETWEEN`, `IN`, `LIKE` и `NULL`
  - Логические условия с операторами `AND`, `OR` и `NOT`
- Правила приоритета операторов, используемых в выражении
- Сортировка строк с использованием предложения `ORDER BY`
- Переменные подстановки
- Команды `DEFINE` и `VERIFY`

# Ограничение числа строк с помощью выборки





## РАБОТНИКИ

	 EMPLOYEE_ID	 LAST_NAME	 JOB_ID	 DEPARTMENT_ID
1	100	King	AD_PRES	90
2	101	Kochhar	AD_VP	90
3	102	De Haan	AD_VP	90
4	103	Hunold	IT_PROG	60
5	104	Ernst	IT_PROG	60
6	107	Lorentz	IT_PROG	60

...

**«извлечь записи  
всех работников  
отдела 90»**



	 EMPLOYEE_ID	 LAST_NAME	 JOB_ID	 DEPARTMENT_ID
1	100	King	AD_PRES	90
2	101	Kochhar	AD_VP	90
3	102	De Haan	AD_VP	90

# Ограничение числа выбранных строк

- Ограничение набора возвращаемых строк с помощью предложения WHERE:

```
SELECT * | { [DISTINCT] столбец | выражение [псевдоним] , ... }  
FROM      таблица  
[WHERE условия] ;
```

- Предложение WHERE следует за предложением FROM.

# Использование предложения WHERE

```
SELECT employee_id, last_name, job_id,  
department_id  
FROM employees  
WHERE department_id = 90;
```

	EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
1	100	King	AD_PRES	90
2	101	Kochhar	AD_VP	90
3	102	De Haan	AD_VP	90

# Символьные строки и даты

- Символьные строки и значения дат заключаются в одиночные кавычки.
- Символьные значения проверяются с учетом регистра, а значения дат – с учетом формата.
- Для вывода дат по умолчанию используется формат 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-FEB-96' ;
```

# Операторы сравнения

Оператор	Значение
=	Равно
>	Больше
>=	Больше или равно
<	Меньше
<=	Меньше или равно
<>	Не равно
BETWEEN ...AND...	Между двумя значениями (включительно)
IN (set)	Совпадает с любым значением из
LIKE	Соответствует шаблону символов
IS NULL	Является значением null



# Использование операторов сравнения

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

	LAST_NAME	SALARY
1	Matos	2600
2	Vargas	2500

# Условия диапазона, использующие оператор BETWEEN

Оператор BETWEEN используется для вывода строк на основе диапазона значений:

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

↑  
Нижний предел

↑  
Верхний предел

	LAST_NAME	SALARY
1	Rajs	3500
2	Davies	3100
3	Matos	2600
4	Vargas	2500

# Условие принадлежности, использующее оператор IN

Оператор IN используется для проверки совпадения со значениями списка:

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

	EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
1	101	Kochhar	17000	100
2	102	De Haan	17000	100
3	124	Mourgos	5800	100
4	149	Zlotkey	10500	100
5	201	Hartstein	13000	100
6	200	Whalen	4400	101
7	205	Higgins	12000	101
8	202	Fay	6000	201

# Сопоставление с шаблоном, использующее оператор LIKE

- Оператор LIKE применяется при поиске допустимых строковых значений с использованием подстановочных знаков.
- Условие поиска может включать символьный или числовой литерал:
  - % обозначает произвольное число символов (в том числе, возможно, нулевое).
  - \_ обозначает строго один символ.

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

# Комбинирование подстановочных знаков

- При сопоставлении с шаблоном можно использовать подстановочные знаки (% , \_) в различных сочетаниях с символами-литералами:

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

	LAST_NAME
1	Kochhar
2	Lorentz
3	Mourgos

- Идентификатор ESCAPE позволяет искать сами символы % и \_.

# Использование условий NULL

Проверка на наличие пустых значений (null) производится с помощью оператора IS NULL.

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

	LAST_NAME	MANAGER_ID
1	King	(null)

# Определение условий с использованием логических операторов

Оператор	Значение
AND	Возвращает TRUE, если <i>оба</i> составляющих условия истинны
OR	Возвращает TRUE, если <i>любое</i> из составляющих условий истинно
NOT	Возвращает TRUE, если условие ложно

# Использование оператора AND

В случае оператора `AND` необходимо, чтобы оба составляющих условия были истинными:

```
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
1	149	Zlotkey	SA_MAN	10500
2	201	Hartstein	MK_MAN	13000



# Использование оператора OR

В случае оператора OR необходимо, чтобы любое из составляющих условий было истинным:

```
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
1	100	King	AD_PRES	24000
2	101	Kochhar	AD_VP	17000
3	102	De Haan	AD_VP	17000
4	124	Mourgos	ST_MAN	5800
5	149	Zlotkey	SA_MAN	10500
6	174	Abel	SA_REP	11000
7	201	Hartstein	MK_MAN	13000
8	205	Higgins	AC_MGR	12000

# Использование оператора NOT

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

	LAST_NAME	JOB_ID
1	De Haan	AD_VP
2	Fay	MK_REP
3	Gietz	AC_ACCOUNT
4	Hartstein	MK_MAN
5	Higgins	AC_MGR
6	King	AD PRES
7	Kochhar	AD_VP
8	Mourgos	ST_MAN
9	Whalen	AD_ASST
10	Zlotkey	SA_MAN

# План занятия

- Средства ограничения числа строк:
  - Предложение `WHERE`
  - Условия сравнения с операторами `=`, `<=`, `BETWEEN`, `IN`, `LIKE` и `NULL`
  - Логические условия с операторами `AND`, `OR` и `NOT`
- Правила приоритета операторов, используемых в выражении
- Сортировка строк с использованием предложения `ORDER BY`
- Переменные подстановки
- Команды `DEFINE` и `VERIFY`

# Правила приоритета

Оператор	Значение
1	Арифметические операторы
2	Оператор конкатенации
3	Условия сравнения
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Не равно
7	Логическое условие NOT
8	Логическое условие AND
9	Логическое условие OR

**Правила приоритета можно переопределять с помощью скобок.**

# Правила приоритета

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

1

	LAST_NAME	JOB_ID	SALARY
1	King	AD_PRES	24000
2	Abel	SA_REP	11000
3	Taylor	SA_REP	8600
4	Grant	SA_REP	7000

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

2

	LAST_NAME	JOB_ID	SALARY
1	King	AD_PRES	24000

# План занятия

- Средства ограничения числа строк:
  - Предложение `WHERE`
  - Условия сравнения с операторами `=`, `<=`, `BETWEEN`, `IN`, `LIKE` и `NULL`
  - Логические условия с операторами `AND`, `OR` и `NOT`
- Правила приоритета операторов, используемых в выражении
- **Сортировка строк с использованием предложения `ORDER BY`**
- Переменные подстановки
- Команды `DEFINE` и `VERIFY`

# Использование предложения ORDER BY

- Сортировка извлеченных строк с помощью предложения ORDER BY:
  - ASC: в порядке возрастания (по умолчанию)
  - DESC: в порядке убывания
- Предложение ORDER BY должно быть последним в инструкции SELECT:

```
SELECT last_name, job_id, department_id,  
hire_date  
FROM employees  
ORDER BY hire_date ;
```

	LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
1	King	AD_PRES	90	17-JUN-87
2	Whalen	AD_ASST	10	17-SEP-87
3	Kochhar	AD_VP	90	21-SEP-89
4	Hunold	IT_PROG	60	03-JAN-90
5	Ernst	IT_PROG	60	21-MAY-91
6	De Haan	AD_VP	90	13-JAN-93

...

# Сортировка

- Сортировка в порядке убывания:

```
SELECT last_name, job_id, department_id,  
hire_date  
FROM employees  
ORDER BY hire_date DESC ;
```

1

- Сортировка по псевдониму столбца:

```
SELECT employee_id, last_name, salary*12 annsal  
FROM employees  
ORDER BY annsal ;
```

2



# Сортировка

- Сортировка по номеру позиции столбца:

```
SELECT last_name, job_id, department_id,  
hire_date  
FROM employees  
ORDER BY 3;
```

3

- Сортировка по нескольким столбцам:

```
SELECT last_name, department_id, salary  
FROM employees  
ORDER BY department_id, salary DESC;
```

4

# План занятия

- Средства ограничения числа строк:
  - Предложение `WHERE`
  - Условия сравнения с операторами `=`, `<=`, `BETWEEN`, `IN`, `LIKE` и `NULL`
  - Логические условия с операторами `AND`, `OR` и `NOT`
- Правила приоритета операторов, используемых в выражении
- Сортировка строк с использованием предложения `ORDER BY`
- **Переменные подстановки**
- Команды `DEFINE` и `VERIFY`

# Переменные подстановки



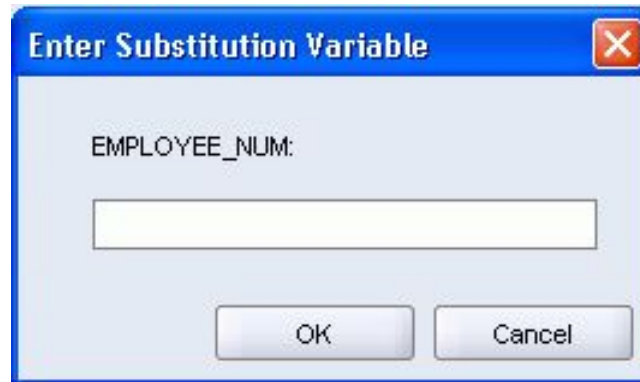
# Переменные подстановки

- Переменные подстановки позволяют:
  - Временно сохранять значения переменных с одним амперсандом (&) или двумя амперсандами (&&)
- Переменные подстановки дополняют следующие элементы:
  - Условия `WHERE`
  - Предложения `ORDER BY`
  - Выражения столбцов
  - Имена таблиц
  - Инструкции `SELECT` в целом

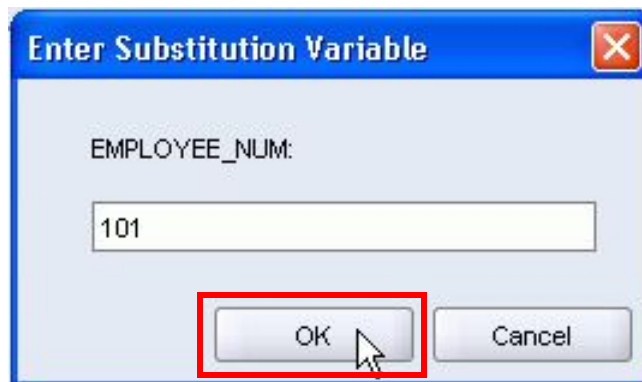
# Использование переменной подстановки с одним амперсандом

Переменная с амперсандом (&) позволяет запрашивать ввод значения пользователем:

```
SELECT employee_id, last_name, salary,  
       department_id  
FROM   employees  
WHERE  employee_id = &employee_num;
```



# Использование переменной подстановки с одним амперсандом



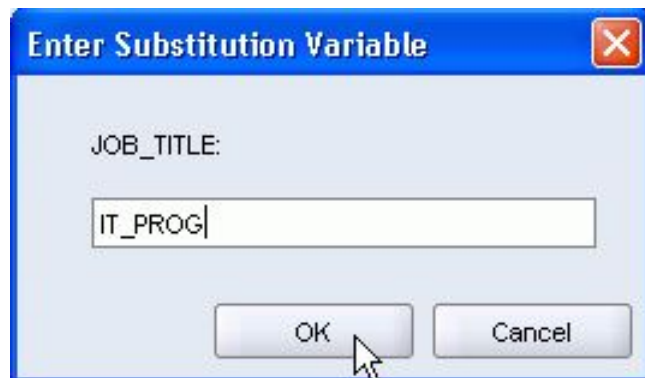
A dialog box titled "Enter Substitution Variable" with a close button (X) in the top right corner. The text "EMPLOYEE\_NUM:" is displayed above a text input field containing the value "101". Below the input field, the "OK" button is highlighted with a red rectangular border, and a mouse cursor is pointing at it. A "Cancel" button is located to the right of the "OK" button.

	EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID
1	101	Kochhar	17000	90

# Символьные значения и даты в переменных подстановки

Используйте одиночные кавычки для дат и символьных значений:

```
SELECT last_name, department_id, salary*12  
FROM employees  
WHERE job_id = '&job_title' ;
```

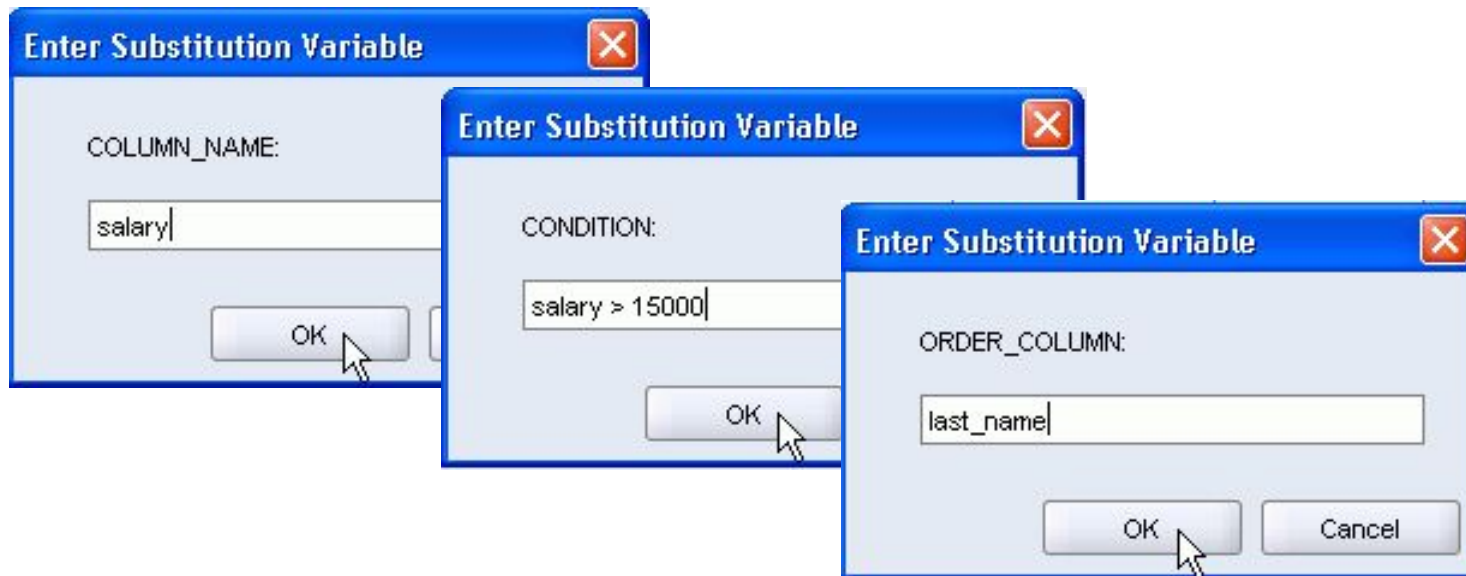


A dialog box titled "Enter Substitution Variable" with a close button (X) in the top right corner. It contains a label "JOB\_TITLE:" followed by a text input field containing "IT\_PROG". At the bottom are "OK" and "Cancel" buttons. A mouse cursor is pointing at the "OK" button.

	LAST_NAME	DEPARTMENT_ID	SALARY*12
1	Hunold	60	108000
2	Ernst	60	72000
3	Lorentz	60	50400

# Задание имен столбцов, выражений и текста

```
SELECT employee_id, last_name, job_id, &column_name  
FROM employees  
WHERE &condition  
ORDER BY &order_column ;
```

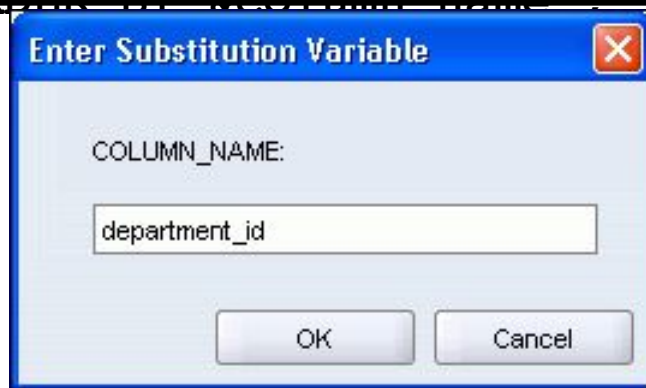




# Использование переменной подстановки с двумя амперсандами

Переменная с двумя амперсандами (&&) позволяет повторно использовать значение переменной, не запрашивая каждый раз ввод пользователя:

```
SELECT employee_id, last_name, job_id,
&&column_name
FROM employees
ORDER BY &column_name;
```



	EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
1	200	Whalen	AD_ASST	10
2	201	Hartstein	MK_MAN	20
3	202	Fay	MK_REP	20

...

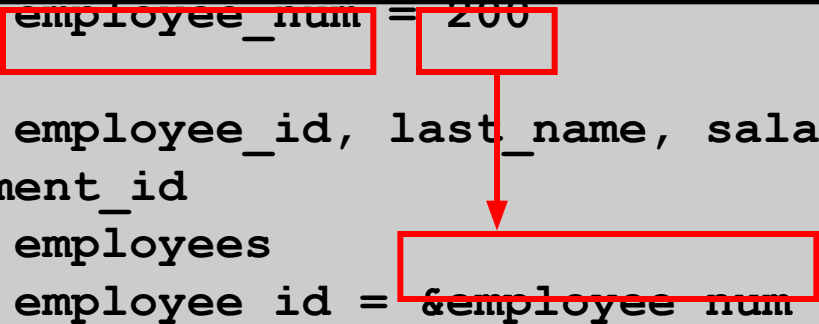
# План занятия

- Средства ограничения числа строк:
  - Предложение `WHERE`
  - Условия сравнения с операторами `=`, `<=`, `BETWEEN`, `IN`, `LIKE` и `NULL`
  - Логические условия с операторами `AND`, `OR` и `NOT`
- Правила приоритета операторов, используемых в выражении
- Сортировка строк с использованием предложения `ORDER BY`
- Переменные подстановки
- Команды `DEFINE` и `VERIFY`

# Использование команды DEFINE

- Команда DEFINE используется для создания переменной и присваивания ей значения.
- Команда UNDEFINE используется для удаления переменной.

```
DEFINE employee_num = 200  
  
SELECT employee_id, last_name, salary,  
department_id  
FROM employees  
WHERE employee_id = &employee_num ;  
  
UNDEFINE employee_num
```

A diagram illustrating the substitution of a variable in a SQL query. A red box highlights the variable 'employee\_num' in the 'DEFINE' statement. Another red box highlights the value '200'. A red arrow points from the '200' to a third red box that highlights '&employee\_num' in the 'WHERE' clause of the 'SELECT' statement, demonstrating how the value is substituted into the query.

# Использование команды VERIFY

Команда VERIFY позволяет переключать режим вывода переменной подстановки до и после того, как та будет заменена фактическим значением в SQL Developer:

```
SET VERIFY ON
```

```
SELECT employee_id, last_name, salary  
FROM   employees  
WHERE  employee_id = &employee_num;
```

Enter Substitution Variable

EMPLOYEE\_NUM:

200

OK Cancel

Results Script Output Explain Autotrace DBMS Output

```
SELECT employee_id, last_name, salary  
FROM   employees  
WHERE  employee_id = 200
```

EMPLOYEE_ID	LAST_NAME	SALARY
200	Whalen	4400

1 rows selected

# Заключение

На этом занятии были изучены следующие темы:

- Ограничение числа выводимых строк с помощью предложения WHERE:
  - Использование условий сравнения
  - Использование операторов BETWEEN, IN, LIKE и NULL
  - Применение логических операторов AND, OR и NOT
- Сортировка выводимых строк с помощью предложения ORDER BY:

```
SELECT * | {[DISTINCT] столбец | выражение  
[псевдоним] , ... }  
FROM      таблица  
[WHERE условия]  
[ORDER BY {столбец, выражение, псевдоним} [ASC|DESC]] ;
```

- Использование переменных подстановки с амперсандами для ограничения и сортировки вывода во время выполнения

## Упражнение 2: обзор

Это упражнение охватывает следующие темы:

- Выбор данных и изменение порядка вывода строк
- Ограничение числа строк с помощью предложения `WHERE`
- Сортировка строк с помощью предложения `ORDER BY`
- Повышение гибкости инструкций SQL `SELECT` с помощью переменных подстановки













