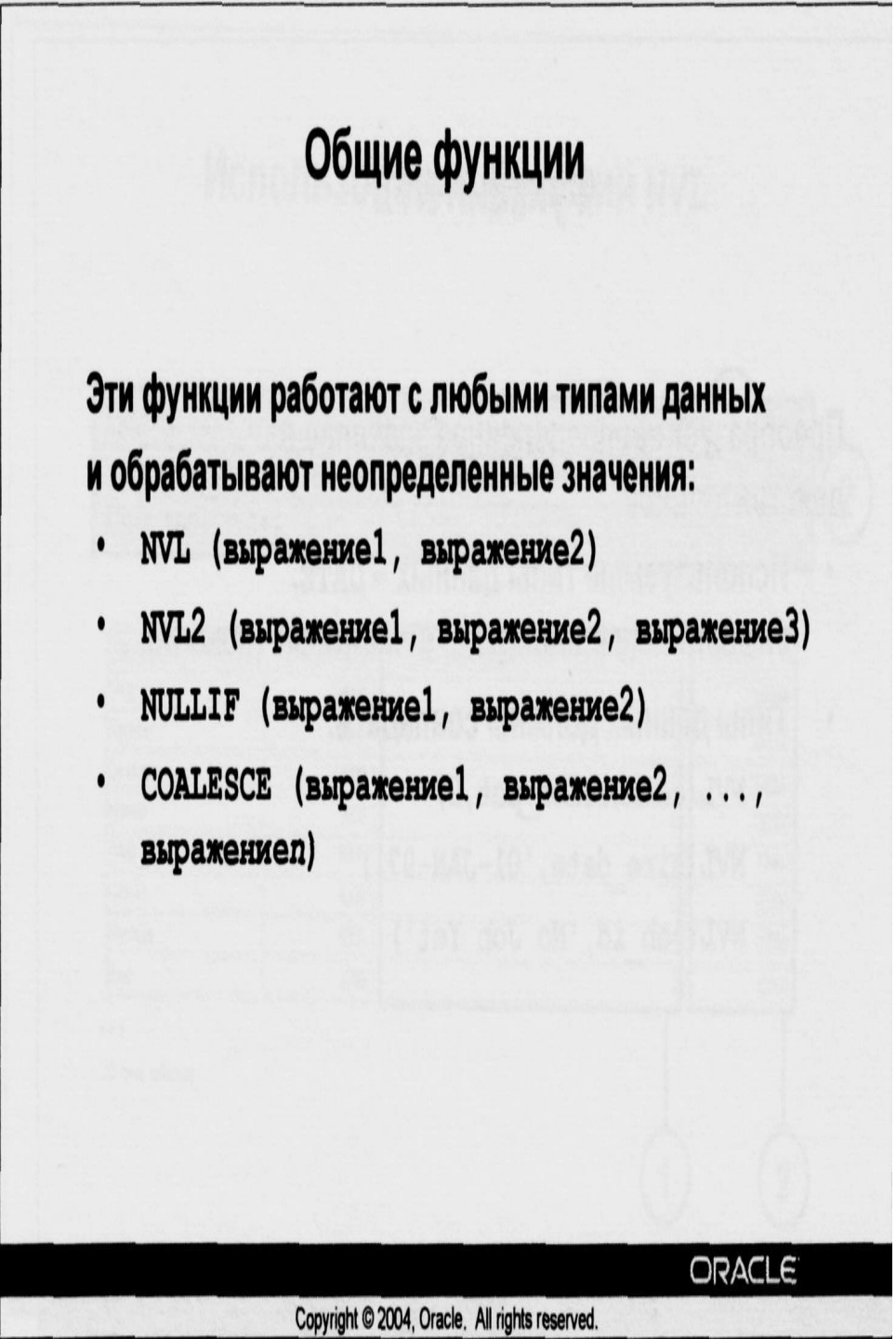
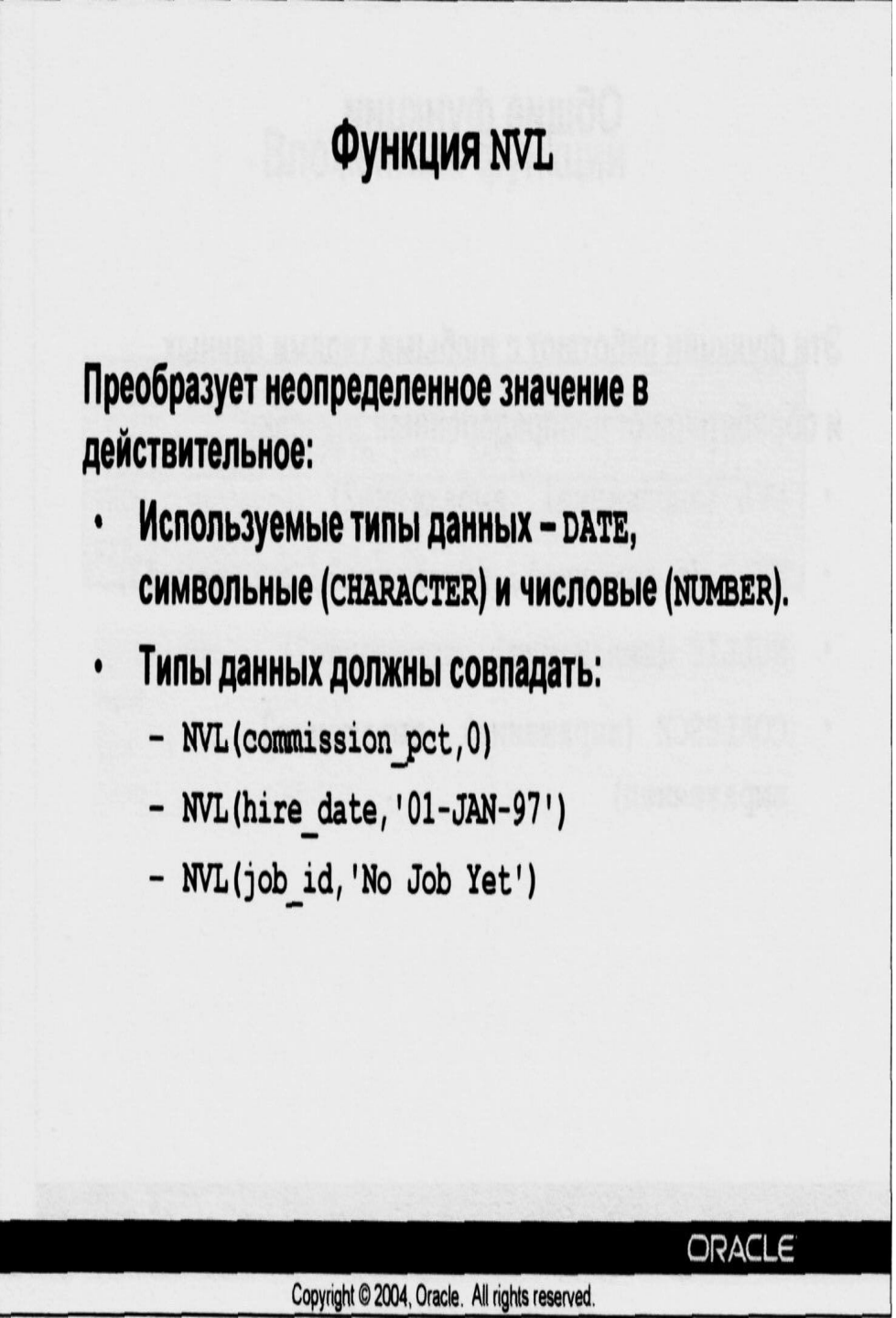


- NVL – преобразует неопределенное значение в действительное
- NVL2 – Если Выражение1 определено (is not null), NVL2 возвратит выражение2. Если Выражение1 не определено (is null), NVL2 возвратит выражение2. Аргумент выражения1 может быть любого типа.
- NULLIF – сравнивает два выражения и возвращает неопределенное значение (Null), если выражения равны, или возвращает первое выражение в противном случае.
- Coalesce – возвращает первое определенное значение из списка выражения.



- Функция NVL используется для преобразования неопределенного значения (NULL) в действительное.

Синтаксис

NVL(выражение1, выражение2)

Выражение1 – исходное значение или выражение, которое может содержать неопределённое значение.

Выражение2 – конечное значение для преобразования неопределенного значения.

NVL – используется для преобразования любого типа, но тип возвращаемых донных всегда такой как у выражения1

NUMBER –

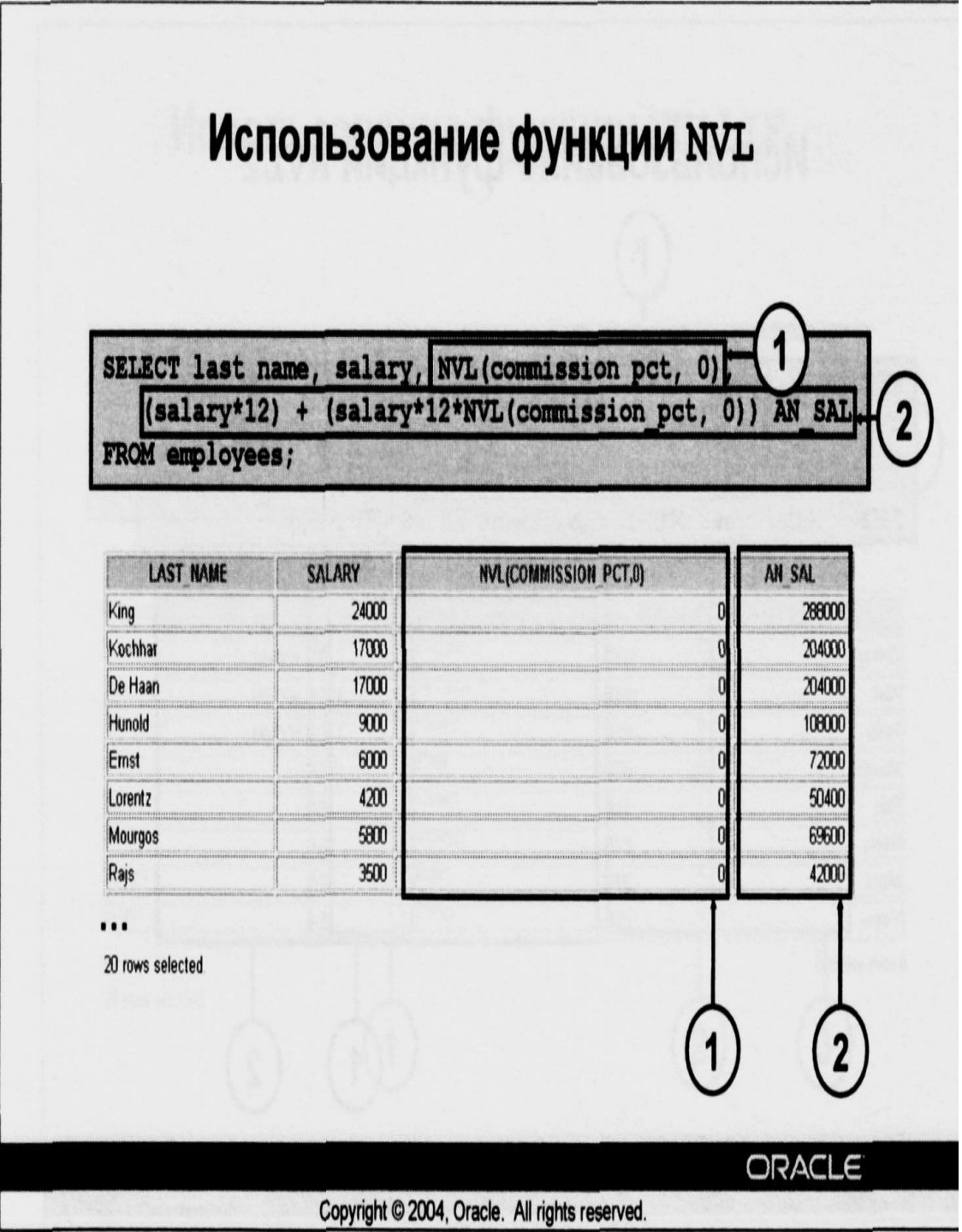
NVL(числовой_столбец,9)

DATE –

NVL(столбец_даты,’01-JAN-95’)

CHAR или VARCHAR2 –

NVL(символьный_столбец, ‘Unavailable’)



Для вычисления годового дохода служащих, необходимо оклад умножить на 12, а затем прибавить сумму комиссионных.

Только для служащих с комиссионными, а у кого не определено значение комиссионных используем NVL.

Использование функции NVL2

```
SELECT last name,  salary, commission pct
       NVL2(commission_pct,
            'SAL+COMM', 'SAL') income
FROM   employees WHERE department id IN (50, 80);
```

LAST_NAME	SALARY	COMMISSION_PCT	INCOME
Zlotkey	10500	.2	SAL+COMM
Abel	11000	.3	SAL+COMM
Taylor	8600	.2	SAL+COMM
Mourgos	5800		SAL
Rajs	3500		SAL
Davies	3100		SAL
Matos	2600		SAL
Vargas	2500		SAL

8 rows selected.

ORACLE

- Функция NVL2 проверяет первое выражение. Если оно определено, тогда функция NVL2 возвращает второе выражение. Если первое выражение не определено, результатом работы будет третье выражение.

NVL2(выражение1,выражение2, выражение3)

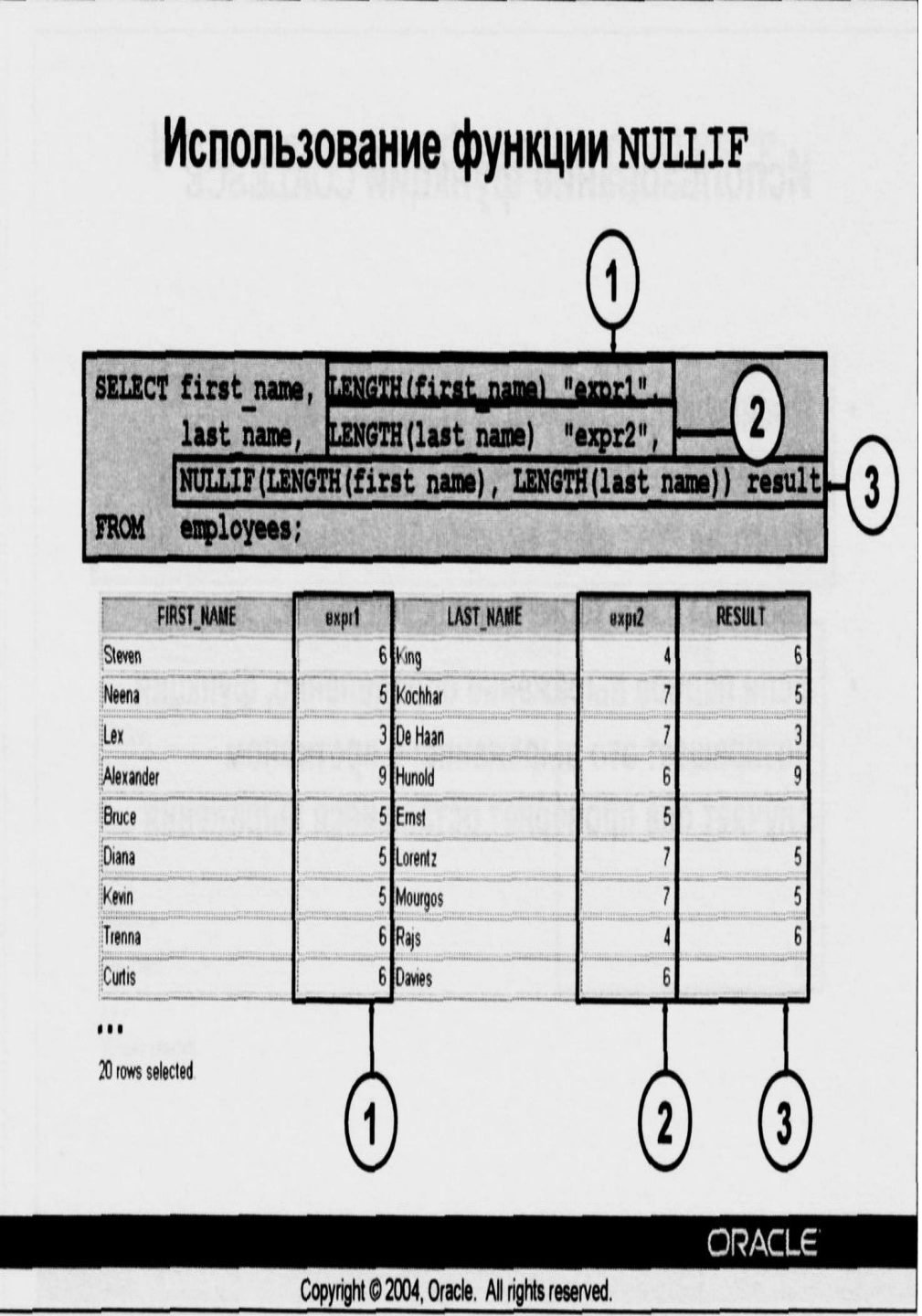
Выражение1 – исходное значение или выражение

Выражение2 – возвращает значение, если выражение1 определено.

Выражение3 – возвращает значение, если выражение не определено.

В примере проверка значения commission_pst. Если есть значение, то ‘sal+comm’. Если нет, то ‘sal’.

Тип выражения 2 и 3 может быть любого типа, кроме LONG. Если типы данных выражения 2 и 3 различные, то сервер выражения3 преобразует в тип выражения2. Здесь преобразование не требуется.



- Функция NULLIF сравнивает два выражения. Если они равны, функция возвращает неопределенное значение, если нет – первое выражение

NULLIF(выражение1, выражение2)

Выражение1 – исходное значение сравниваемое с выражением2.

Выражение2 – исходное значение сравниваемое с выражением1.

Использование функции COALESCE

- Преимущество функции COALESCE по сравнению с функцией NVL состоит в том, что функция COALESCE может обрабатывать несколько альтернативных значений.
- Если первое выражение определенно, функция возвращает это выражение; в противном случает она проверяет оставшиеся выражения

ORACLE

- Функция COALESCE возвращает первое определенное выражение в списке.

COALESCE(выражение1, выражение2,…выражениеN)

Выражение1- возвращаемое выражение, если оно имеет определенное значение.

Выражение2 – возвращаемое выражение, если первое выражение не определено, а это выражение имеет определенное значение.

выражениеN – возвращаемое выражение, если предыдущее выражение не определено.

Использование функции COALESCE

```
SELECT last_name,
       COALESCE(manager_id,commission_pct, -1) comm
FROM   employees
ORDER BY commission_pct;
```

LAST_NAME	COMM
Grant	149
Zlotkey	100
Taylor	149
Abel	149
King	-1
Kochhar	100
De Haan	100

...

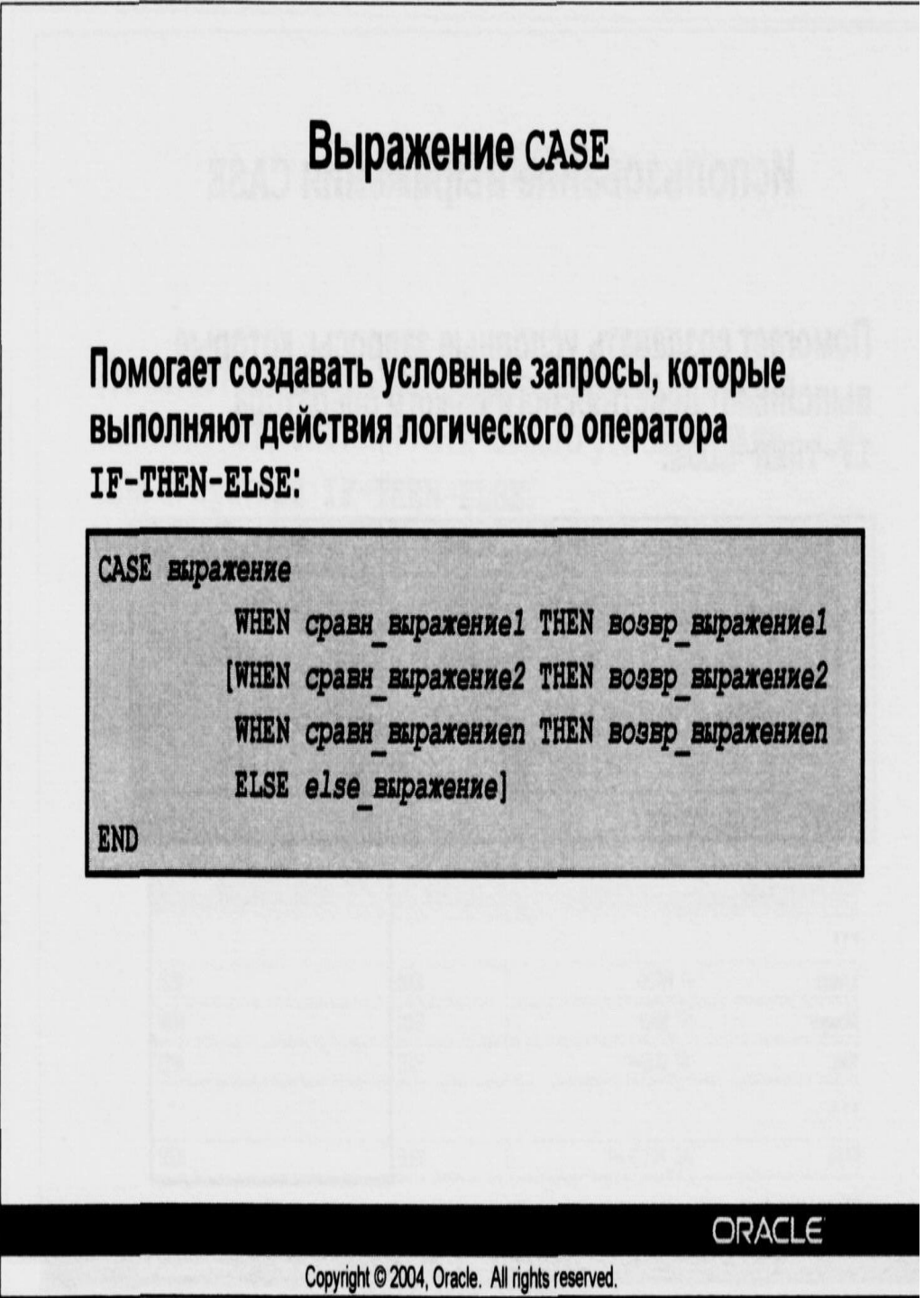
20 rows selected.

ORACLE

- В примере выводится табельный номер менеджера, если он определен. Если табельный номер менеджера не задан выводится commission_pst, если они определены. Если ни табельный номер менеджера, ни комиссионные не установлены, выводится -1.

Условные выражения

- Позволяют применять логические конструкции ЕСЛИ-ТО-ИНАЧЕ (IF-THEN-ELSE) внутри команды SQL
- Два метода:
  - выражение CASE
  - функция DECODE



Выражение CASE позволяет производить логическую обработку оператора IF-THEN-ELSE в командах SQL, не вызывая процедуры.

В простом выражении CASE сервер Oracle ищет первую пару WHEN … THEN, в которой совпадают выражение и сравн_выражение и возвращает возвр_выражение. Если нет совпадений ни в одном из сравниваемых пар WHEN… THEN и есть предложение ELSE, возвращается else_выражение. Если такого предложения ELSE нет, возвращается неопределённое значение. Нельзя задавать литерал NULL для возвр_выражений и else_выражений. Все возвращаемые значения должны быть одного типа.

Использование выражения CASE

Помогает создавать условные запросы, которые выполняют действия логического оператора IF-THEN-ELSE:

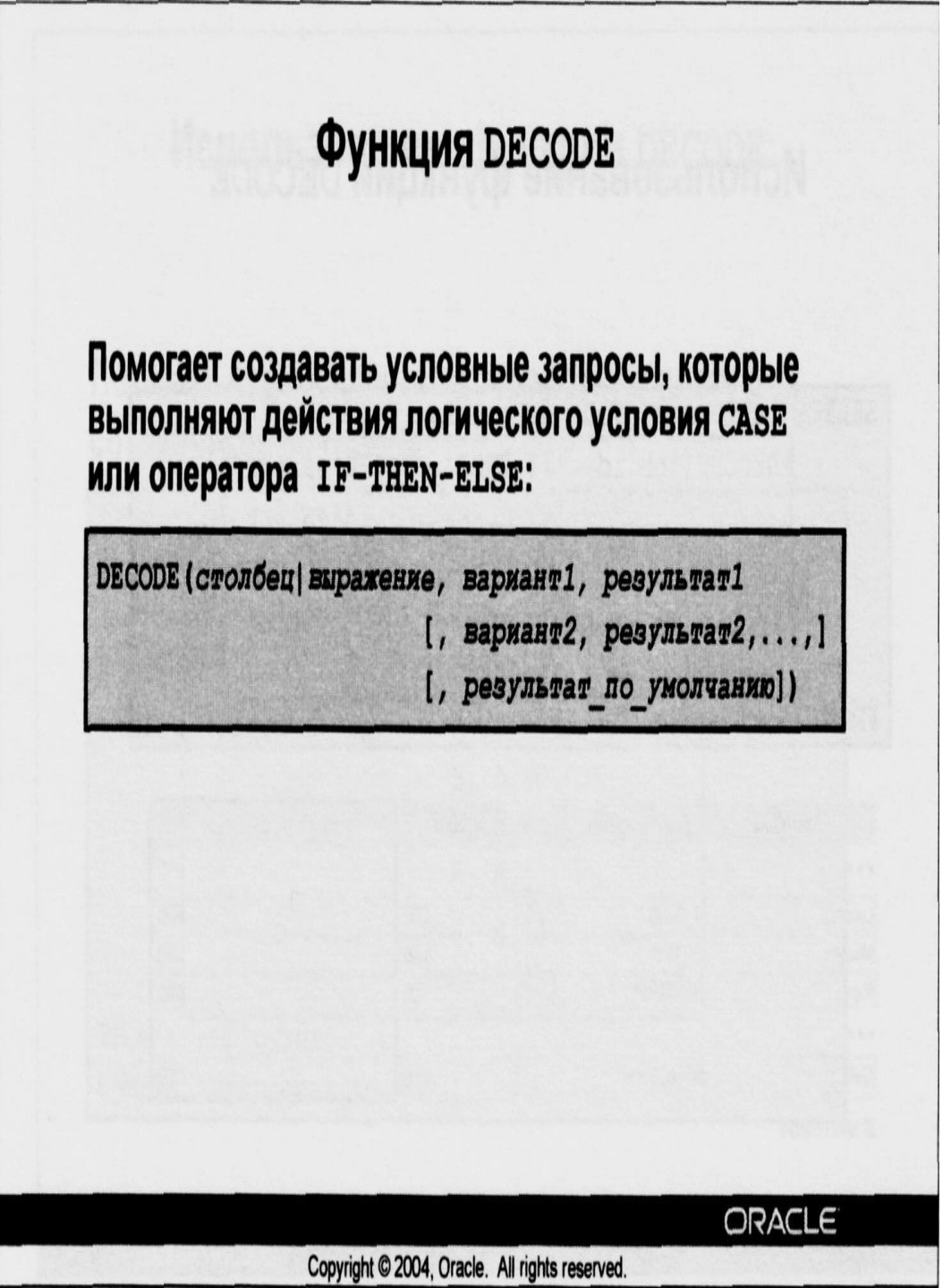
```
SELECT last_name, job_id, salary,
       CASE job_id WHEN 'IT_PROG'  THEN  1.10*salary
                   WHEN 'ST_CLERK' THEN  1.15*salary
                   WHEN 'SA_REP'   THEN  1.20*salary
       ELSE       salary END     "REVISED_SALARY"
FROM   employees;
```

LAST_NAME	JOB_ID	SALARY	REVISED_SALARY
...			
Lorentz	IT_PROG	4200	4620
Mourgos	ST_MAN	5800	5800
Rajs	ST_CLERK	3500	4025
...			
Gietz	AC_ACCOUNT	8300	8300

20 rows selected.

ORACLE

- В приведенной команде SQL расшифровывается значение идентификатора должности JOB_ID. Если знечение JOB_ID совпадает с IT_PROG, оклад повышается на 10%; если JOB_ID равно ST_CLERK, повышение на 15%; если JOB_ID равно SA_REP, оклад увеличивается на 20%. Для всех остальных должностной оклад не изменяется.
- Такую же команду можно написать с помощью функции DECODE.



- Функция DECODE действует подобно IF-THEN-ELSE в различных языках. Функция DECODE расшифровывает столбец или выражение после сравнения его с каждым искомым значением варианта. Если выражение равно искомому значению, функция возвращает соответствующий результат.

Использование функции DECODE

```
SELECT last_name, job_id, salary,
       DECODE(job_id, 'IT_PROG',  1.10*salary,
                      'ST_CLERK', 1.15*salary,
                      'SA_REP',   1.20*salary,
              salary)
       REVISED_SALARY
FROM   employees;
```

LAST_NAME	JOB_ID	SALARY	REVISED_SALARY
...			
Lorentz	IT_PROG	4200	4620
Mourgos	ST_MAN	5800	5800
Rajs	ST_CLERK	3500	4025
...			
Gietz	AC_ACCOUNT	8300	8300

20 rows selected.

Использование функции DECODE

Показать ставку налога на заработную плату для сотрудников 80 отдела:

```
SELECT last_name, salary,
       DECODE (TRUNC(salary/2000, 0),
                          0, 0.00,
                          1, 0.09,
                          2, 0.20,
                          3, 0.30,
                          4, 0.40,
                          5, 0.42,
                          6, 0.44,
                             0.45) TAX_RATE
FROM   employees
WHERE  department_id = 80;
```

ORACLE

Итоги

С помощью функций осуществляются:

- Вычисления с данными
- Изменение отдельных элементов данных
- Манипулирование выводом групп строк
- Изменение форматов дат для вывода
- Преобразование формата данных столбцов
- Обработка неопределенных значений
- Логическая обработка IF-THEN-ELSE

Обзор практического занятия 3 , часть 2

- Составление запросов, требующих использования числовых, символьных функций и функций для работы с датами.
- Использование конкатенации с функциями.
- Составление запросов, нечувствительных к регистру символов, для проверки полезности символьных функций.
- Вычисление продолжительности работы служащего в месяцах и годах.
- Определение даты аттестации служащего