



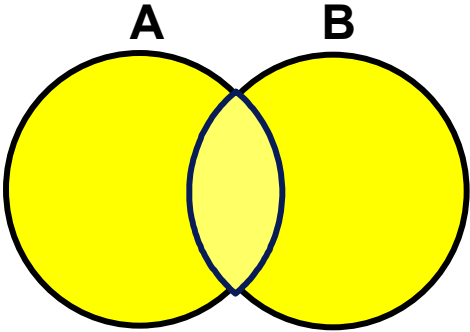
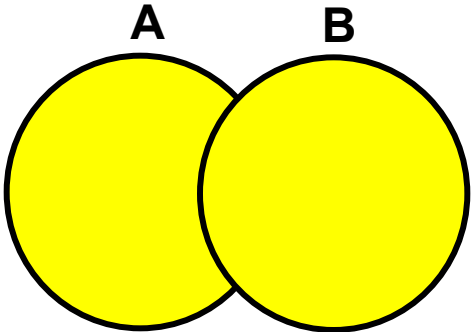
# Использование Set операторов

# Цели занятия

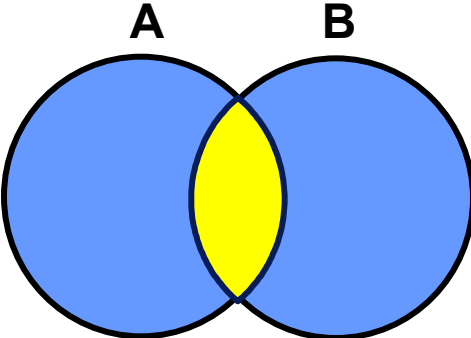
**Чему мы научимся на данном занятии:**

- **Характеризировать set операторы**
- **Использовать set операторы для объединения нескольких запросов в один запрос.**
- **Контролировать порядок возвращаемых строк.**

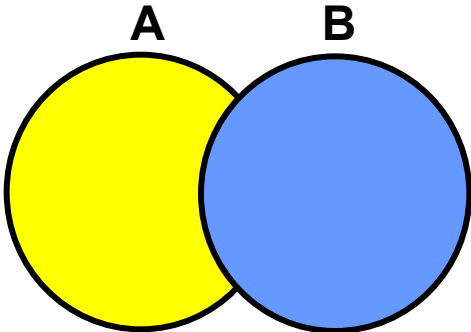
# Set операторы



UNION/UNION ALL



INTERSECT

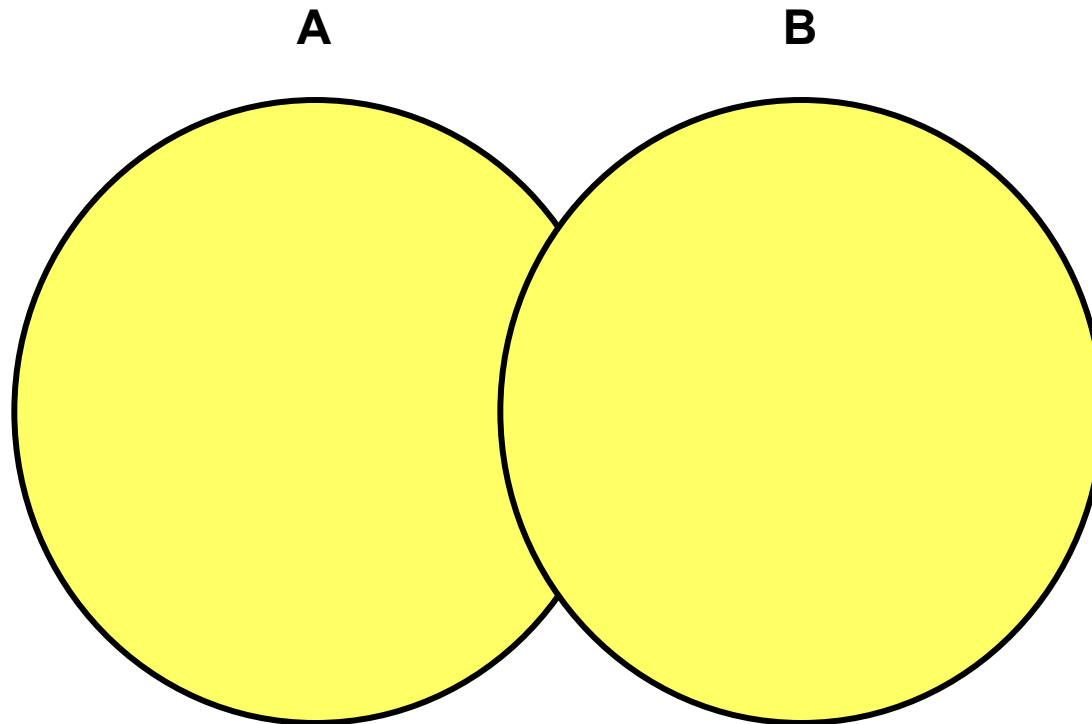


MINUS

# Таблицы, используемых в этом уроке.

- **EMPLOYEES:** Содержит подробную информацию обо всех действующих сотрудниках.
- **JOB\_HISTORY:** Хранит записи о дате начала и дате окончания работы, идентификационные номера сотрудника и отдела, дату смены сотрудником рабочего места.

# Оператор UNION



**Оператор UNION возвращает результаты из обоих запросов, после устранения дублирования.**

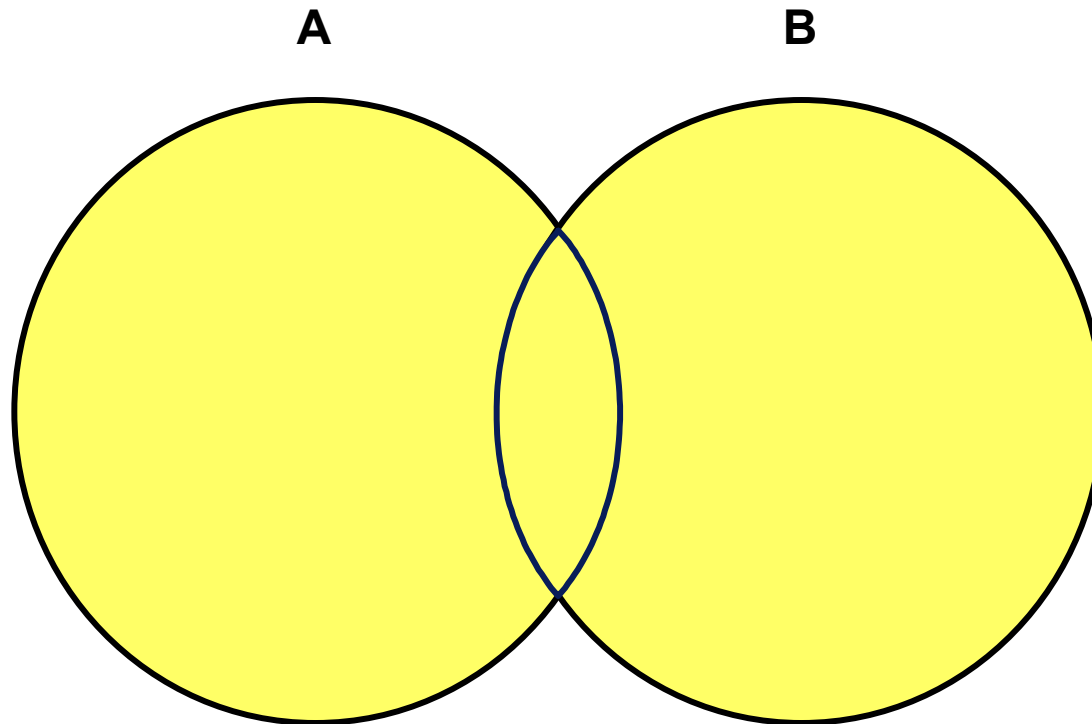
# Использование оператора UNION

Отображаем текущее и предыдущее место работы всех сотрудников. Выводим информацию по каждому сотруднику только один раз.

```
SELECT employee_id, job_id
FROM employees
UNION
SELECT employee_id, job_id
FROM job_history;
```

EMPLOYEE_ID	JOB_ID
100	AD_PRES
101	AC_ACCOUNT
...	
200	AC_ACCOUNT
200	AD_ASST
...	
205	AC_MGR
206	AC_ACCOUNT

# Оператор UNION ALL



**Оператор UNION ALL возвращает результаты из обоих запросов, в том числе все дубли.**

# Использование оператора UNION ALL

Отображение текущих и предыдущих отделов всех сотрудников.

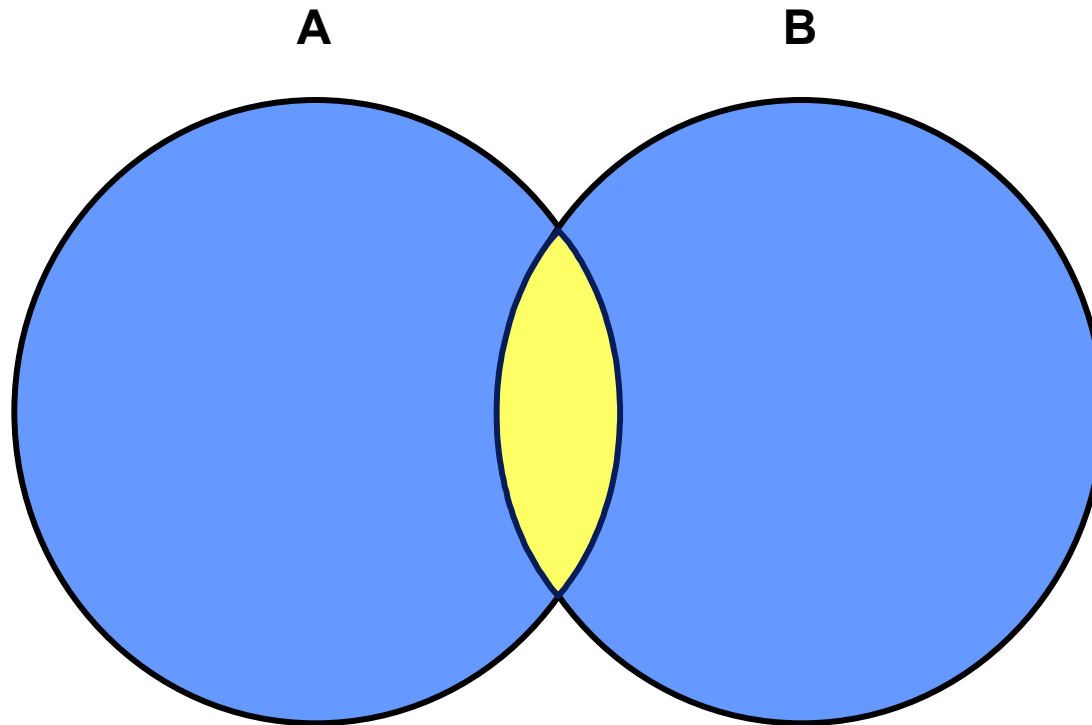
```
SELECT employee_id, job_id, department_id
FROM employees
UNION ALL
SELECT employee_id, job_id, department_id
FROM job_history
ORDER BY employee_id;
```

EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID
100	AD_PRES	90
101	AD_VP	90
...		
200	AD_ASST	10
200	AD_ASST	90
200	AC_ACCOUNT	90
...		
205	AC_MGR	110
206	AC_ACCOUNT	110

30 rows selected.



# Оператор INTERSECT



**Оператор INTERSECT возвращает строки, которые являются общими для обоих запросов.**

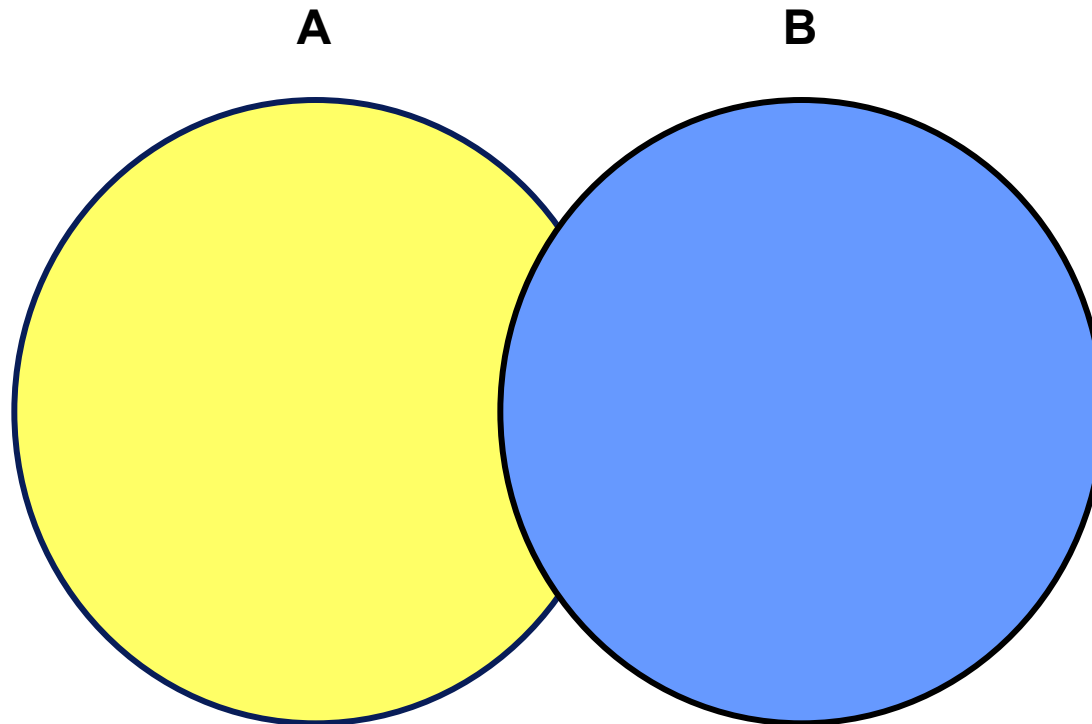
# Использование оператора INTERSECT

Выводим идентификаторы сотрудников и идентификаторы их места работы, которые в настоящее время работают на должности, на которую они были первоначально наняли (то есть, они меняли рабочее места, но теперь вернулись).

```
SELECT employee_id, job_id
FROM employees
INTERSECT
SELECT employee_id, job_id
FROM job_history;
```

EMPLOYEE_ID	JOB_ID
176	SA_REP
200	AD_ASST

# MINUS Operator



**Оператор MINUS возвращает строки из первого запроса, которые не присутствуют во втором.**

# Оператор MINUS

Выведем идентификаторы тех сотрудников, которые не меняли своё рабочее место ни разу.

```
SELECT employee_id,job_id
FROM employees
MINUS
SELECT employee_id,job_id
FROM job_history;
```

EMPLOYEE_ID	JOB_ID
100	AD_PRES
101	AD_VP
102	AD_VP
103	IT_PROG
...	
201	MK_MAN
202	MK_REP
205	AC_MGR
206	AC_ACCOUNT

18 rows selected.

# Рекомендации по использованию Set операторов

- **Выражения в clause SELECT должны совпадать по количеству столбцов и типу их данных.**
- **Для clause ORDER BY:**
  - **Может использоваться только в самом конце выражения**
  - **Будет использовать название столбца или его псевдоним из первого SELECT оператора.**

# Сервер Oracle и Set операторы

- Повторяющиеся строки автоматически устраняются, за исключением UNION ALL.
- Имена столбцов из первого запроса появляются в результате.
- Результаты по умолчанию будут отсортированы в порядке возрастания, за исключением UNION ALL.

# Сопоставление значений

Используя оператор UNION, выведем ID департамента, место его расположения и дату приёма всех сотрудников.

```
SELECT department_id, TO_NUMBER(null)
       location, hire_date
FROM   employees
UNION
SELECT department_id, location_id,  TO_DATE(null)
FROM   departments;
```

DEPARTMENT_ID	LOCATION	HIRE_DATE
10	1700	
10		17-SEP-87
20	1800	
20		17-FEB-96
...		
110	1700	
110		07-JUN-94
190	1700	
		24-MAY-99

27 rows selected.

# Управление порядком вывода строк

Вывести предложение, используя два оператора UNION.

```
COLUMN a_dummy NOPRINT
SELECT 'sing' AS "My dream", 3 a_dummy
FROM dual
UNION
SELECT 'I'd like to teach', 1 a_dummy
FROM dual
UNION
SELECT 'the world to', 2 a_dummy
FROM dual
ORDER BY a_dummy;
```

My dream
I'd like to teach
the world to
sing



# Итог

Чему мы научились из данной лекции :

- Использование **UNION** позволяет получить все записи без дублей (уникальные).
- Использование **UNION ALL** позволяет получить все записи, включая дублирующие.
- Использование **INTERSECT** позволяет получить все записи, которые есть в обоих запросах.
- Использование **MINUS** позволяет получить все уникальные записи, которые есть в первом запросе, но нет во втором.
- Использование **ORDER BY** возможно только в самом конце выражения

# Практика: План

На данной практике будут охвачены следующие темы :

- Использование оператора UNION
- Использование оператора INTERSECT
- Использование оператора MINUS

# Задания

- 1) Вывести числа 1,2,3 в 3 строки.
- 2) Вывести все EMPLOYEE\_ID из таблиц EMPLOYEES и JOB\_HISTORY, с сортировкой по убыванию.
- 3) Вывести все EMPLOYEE\_ID, которые есть в обеих таблицах EMPLOYEES и JOB\_HISTORY.
- 4) Вывести все EMPLOYEE\_ID, которые есть в таблице EMPLOYEES, но нет в JOB\_HISTORY.