

☒ Autocommit    Display  ▼

```
SELECT department_id,  
       MIN(salary) AS min_salary,  
       MAX(salary) AS max_salary  
FROM employees  
GROUP BY department_id;
```

**Results**   Explain   Describe   Saved SQL   History

DEPARTMENT_ID	MIN_SALARY	MAX_SALARY
-	7000	7000
100	6900	12000
30	2500	11000
90	17000	24000
20	6000	13000
70	10000	10000
110	8300	12000
50	2100	8200
80	6100	14000

User: HR

Home > SQL > SQL Commands

☒ Autocommit    Display 10

Save

Run

```
SELECT e.first_name || ' ' || e.last_name AS employee_name,  
       j.job_title  
FROM employees e  
JOIN jobs j  
  ON e.job_id = j.job_id  
WHERE e.salary < (  
       SELECT AVG(salary) FROM employees  
);
```

Results   Explain   Describe   Saved SQL   History

EMPLOYEE_NAME	JOB_TITLE
Bruce Ernst	Programmer
David Austin	Programmer
Valli Pataballa	Programmer
Diana Lorentz	Programmer
Alexander Khoo	Purchasing Clerk
Shelli Baida	Purchasing Clerk
Sigal Tobias	Purchasing Clerk
Guy Himuro	Purchasing Clerk
Karen Colman	Purchasing Clerk

User: HR

[Home](#) > [SQL](#) > [SQL Commands](#)☒ Autocommit Display  ▼

Save

Run

```
SELECT AVG(salary) FROM employees  
);  
  
CREATE OR REPLACE VIEW emp_2002_2005 AS  
SELECT first_name,  
       hire_date  
FROM employees  
WHERE hire_date BETWEEN DATE '2002-01-01' AND DATE '2005-12-31';  
  
SELECT * FROM emp_2002_2005;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

View created.

0.08 seconds

☒ Autocommit   Display  ▼

```
CREATE OR REPLACE VIEW emp_2002_2005 AS
SELECT first_name,
       hire_date
FROM employees
WHERE hire_date BETWEEN DATE '2002-01-01' AND DATE '2005-12-31';

SELECT * FROM emp_2002_2005;
```

**Results**   Explain   Describe   Saved SQL   History

no data found

User: HR

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
CREATE OR REPLACE FUNCTION fact(n IN PLS_INTEGER)
RETURN PLS_INTEGER
IS
    res PLS_INTEGER := 1;
    i   PLS_INTEGER;
BEGIN
    IF n < 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Negative not allowed');
    END IF;

    FOR i IN 1 .. n LOOP
        res := res * i;
    END LOOP;
END;
```

**Results** Explain Describe Saved SQL History

Function created.

0.03 seconds

User: HR

Home &gt; SQL &gt; SQL Commands

☒ Autocommit   Display 10 ▼

```
DECLARE
  v_num PLS_INTEGER := 5;
  v_fact PLS_INTEGER;
BEGIN
  v_fact := fact(v_num);
  DBMS_OUTPUT.PUT_LINE('Factorial is ' || v_fact);
END;
/
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Factorial is 120

Statement processed.

0.00 seconds

```
CREATE OR REPLACE TRIGGER before_delete_trigger  
BEFORE DELETE ON employees -- or any table name as required  
BEGIN  
    DBMS_OUTPUT.PUT_LINE('table updated successfully');  
END;  
/
```

---

**Results** Explain Describe Saved SQL History

---

Trigger created.

0.04 seconds