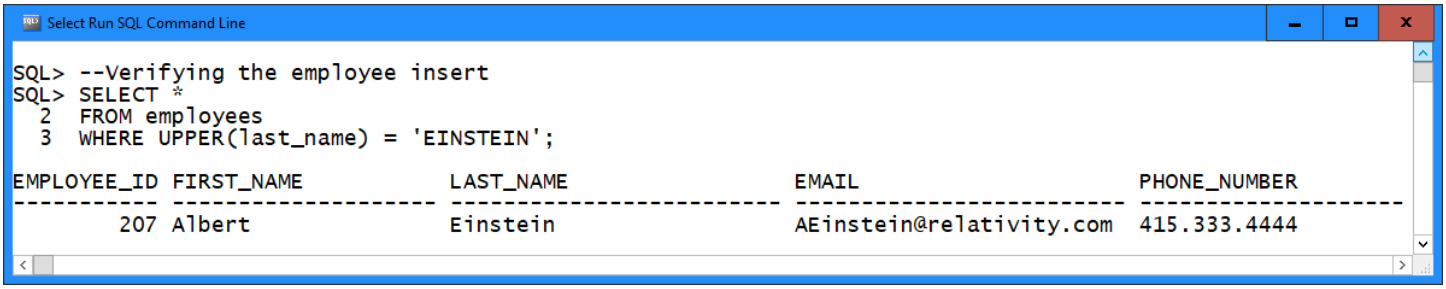
Problem 1:

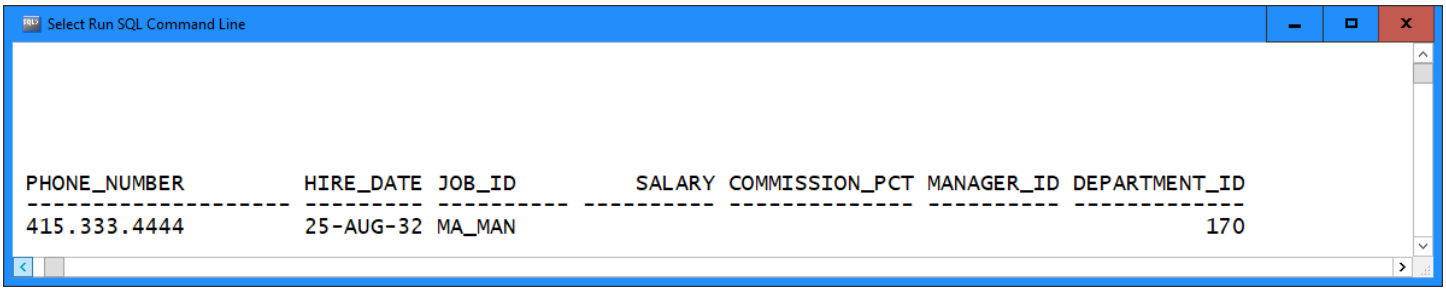
```
--Finding jobs having 'MANAG' in column job_title
SELECT * FROM jobs
WHERE UPPER(job_title) LIKE '%MANAG%';
--No manufacturing manager found, hence need to insert
--Use 'MA_MAN' as job_code based on existing naming convention
--Einstein is pricelesss, therefore no min_salary or max_salary
INSERT INTO jobs
VALUES('MA_MAN','Manufacturing Manager',NULL,NULL);
--Verify new entry in table jobs
SELECT * FROM jobs
WHERE job_id = 'MA_MAN';
--Verify whether department 'Manufacturing' exists
SELECT department_id, department_name
FROM departments
WHERE UPPER(department_name) LIKE '%MANUF%';
--Department 'Manufacturing' exists (ID=170)
--Use 'MA_MAN' job_id for insert into table employees
--Use 'MAX(employee_ID) + 1' as next employee_ID
INSERT INTO employees
VALUES((SELECT MAX(employee_id)+1 FROM employees),'Albert','Einstein',
'AEinstein@relativity.com','415.333.4444','25-AUG-1932','MA_MAN',
NULL,NULL,NULL,(SELECT department_id FROM departments
WHERE UPPER(department_name) = 'MANUFACTURING'));
```

Run SQL Command Line

```
SQL> --Finding jobs having 'MANAG' in column job_title
SQL> SELECT * FROM jobs
  2  WHERE UPPER(job_title) LIKE '%MANAG%';

JOB_ID     JOB_TITLE                          MIN_SALARY MAX_SALARY
---------- ---------------------------------- ---------- ----------
FI_MGR     Finance Manager                          8200      16000
AC_MGR     Accounting Manager                       8200      16000
SA_MAN     Sales Manager                           10000      20000
PU_MAN     Purchasing Manager                       8000      15000
ST_MAN     Stock Manager                            5500       8500
MK_MAN     Marketing Manager                        9000      15000

6 rows selected.

SQL> --No manufacturing manager found, hence need to insert
SQL> --Use 'MA_MAN' as job_code based on existing naming convention
SQL> --Einstein is pricelesss, therefore no min_salary or max_salary
SQL> INSERT INTO jobs
  2  VALUES('MA_MAN','Manufacturing Manager',NULL,NULL);

1 row created.

SQL> --Verify new entry in table jobs
SQL> SELECT * FROM jobs
  2  WHERE job_id = 'MA_MAN';

JOB_ID     JOB_TITLE                          MIN_SALARY MAX_SALARY
---------- ---------------------------------- ---------- ----------
MA_MAN     Manufacturing Manager

SQL> --Verify whether department 'Manufacturing' exists
SQL> SELECT department_id, department_name
  2  FROM departments
  3  WHERE UPPER(department_name) LIKE '%MANUF%';

DEPARTMENT_ID DEPARTMENT_NAME
------------- ------------------------------
          170 Manufacturing

SQL> --Department 'Manufacturing' exists (ID=170)
SQL> --Use 'MA_MAN' job_id for insert into table employees
SQL> --Use 'MAX(employee_ID) + 1' as next employee_ID
SQL> INSERT INTO employees
  2  VALUES((SELECT MAX(employee_id)+1 FROM employees),'Albert','Einstein',
  3  'AEinstein@relativity.com','415.333.4444','25-AUG-1932','MA_MAN',
  4  NULL,NULL,NULL,(SELECT department_id FROM departments
  5  WHERE UPPER(department_name) = 'MANUFACTURING'));

1 row created.
```

```
--Verifying the employee insert
SELECT *
FROM employees
WHERE UPPER(last_name) = 'EINSTEIN';
```





Problem 2:

```
SELECT jh.*, j.min_salary FROM job_history jh
INNER JOIN jobs j ON j.job_id = jh.job_id
ORDER BY jh.employee_id, jh.job_id;

DELETE FROM job_history
WHERE job_id IN
(
        SELECT job_id
        FROM jobs
        WHERE min_salary >= 10000
);

SELECT jh.*, j.min_salary FROM job_history jh
INNER JOIN jobs j ON j.job_id = jh.job_id
ORDER BY jh.employee_id, jh.job_id;
```

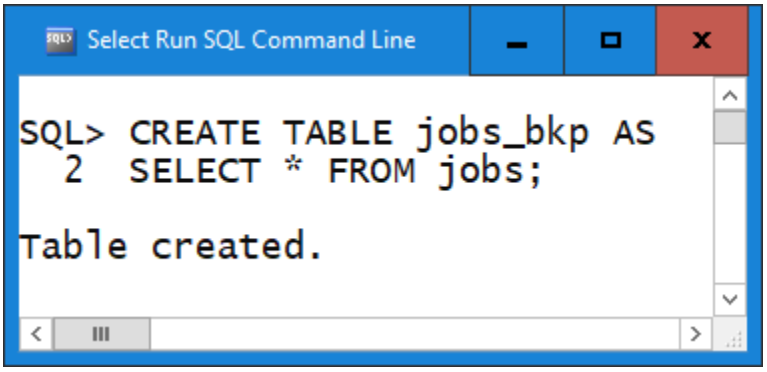
Run SQL Command Line

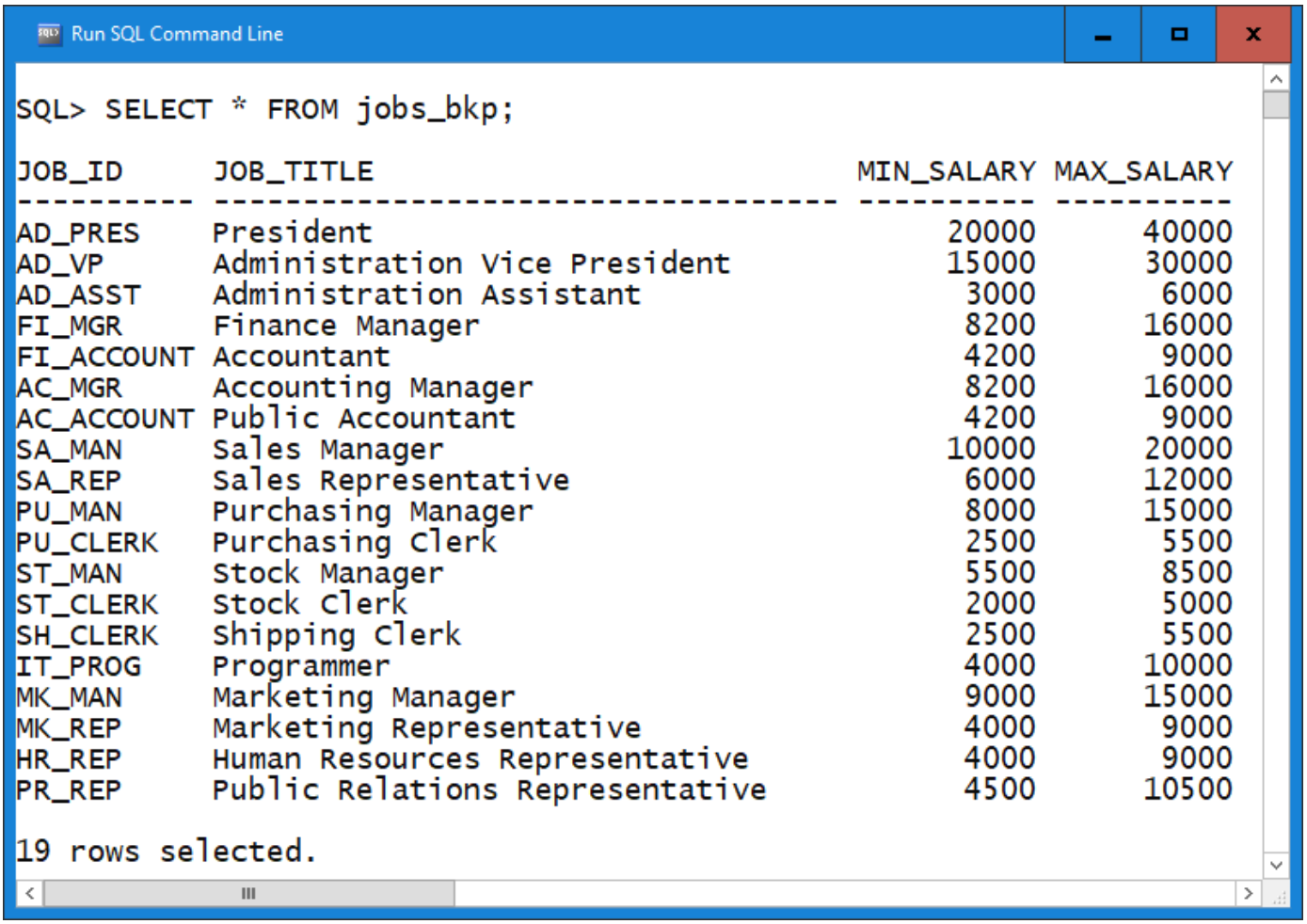
```
SQL> SELECT jh.*, j.min_salary FROM job_history jh
  2  INNER JOIN jobs j ON j.job_id = jh.job_id
  3  ORDER BY jh.employee_id, jh.job_id;

EMPLOYEE_ID START_DAT END_DATE  JOB_ID     DEPARTMENT_ID MIN_SALARY
----------- --------- --------- ---------- ------------- ----------
        101 21-SEP-89 27-OCT-93 AC_ACCOUNT           110       4200
        101 28-OCT-93 15-MAR-97 AC_MGR               110       8200
        102 13-JAN-93 24-JUL-98 IT_PROG               60       4000
        114 24-MAR-98 31-DEC-99 ST_CLERK              50       2000
        122 01-JAN-99 31-DEC-99 ST_CLERK              50       2000
        176 01-JAN-99 31-DEC-99 SA_MAN                80      10000
        176 24-MAR-98 31-DEC-98 SA_REP                80       6000
        200 01-JUL-94 31-DEC-98 AC_ACCOUNT            90       4200
        200 17-SEP-87 17-JUN-93 AD_ASST               90       3000
        201 17-FEB-96 19-DEC-99 MK_REP                20       4000

10 rows selected.

SQL>
SQL> DELETE FROM job_history
  2  WHERE job_id IN
  3  (
  4     SELECT job_id
  5     FROM jobs
  6     WHERE min_salary >= 10000
  7  );

1 row deleted.

SQL>
SQL> SELECT jh.*, j.min_salary FROM job_history jh
  2  INNER JOIN jobs j ON j.job_id = jh.job_id
  3  ORDER BY jh.employee_id, jh.job_id;

EMPLOYEE_ID START_DAT END_DATE  JOB_ID     DEPARTMENT_ID MIN_SALARY
----------- --------- --------- ---------- ------------- ----------
        101 21-SEP-89 27-OCT-93 AC_ACCOUNT           110       4200
        101 28-OCT-93 15-MAR-97 AC_MGR               110       8200
        102 13-JAN-93 24-JUL-98 IT_PROG               60       4000
        114 24-MAR-98 31-DEC-99 ST_CLERK              50       2000
        122 01-JAN-99 31-DEC-99 ST_CLERK              50       2000
        176 24-MAR-98 31-DEC-98 SA_REP                80       6000
        200 01-JUL-94 31-DEC-98 AC_ACCOUNT            90       4200
        200 17-SEP-87 17-JUN-93 AD_ASST               90       3000
        201 17-FEB-96 19-DEC-99 MK_REP                20       4000

9 rows selected.
```

Problem 3:

Part 1:

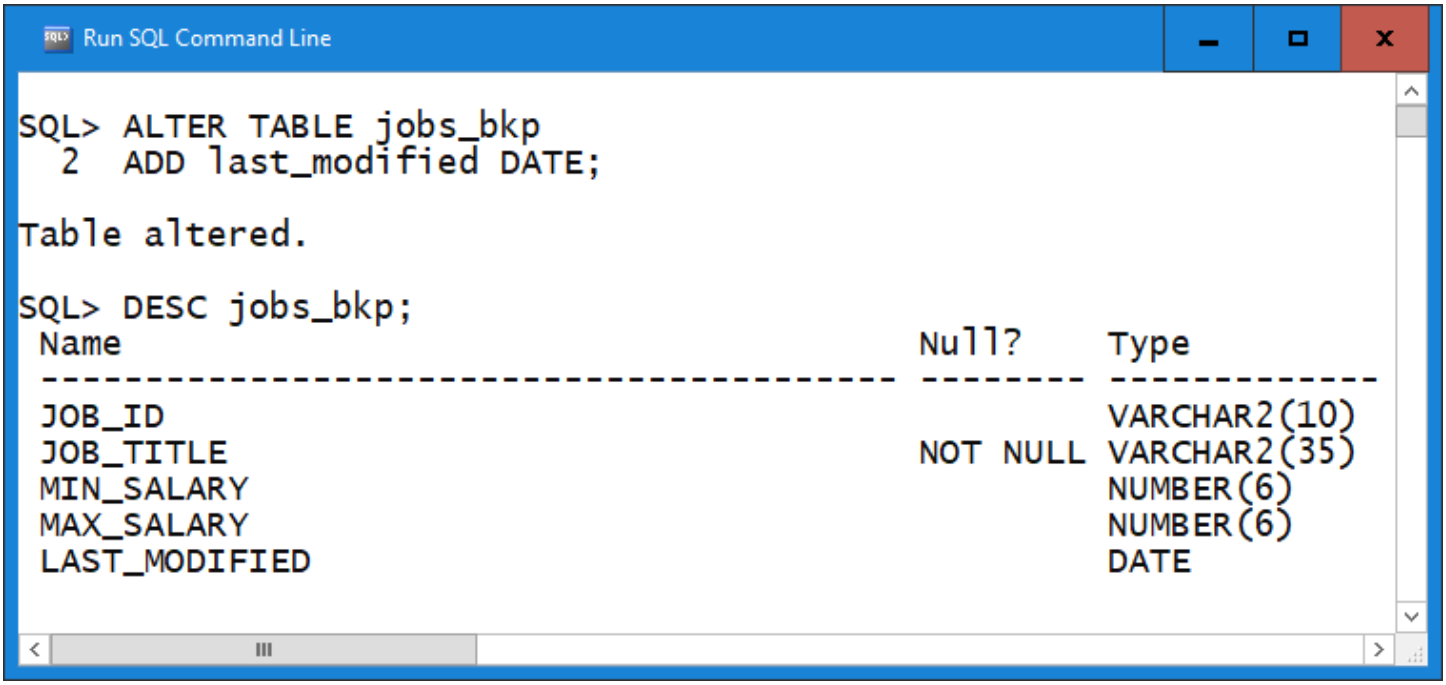
CREATE TABLE jobs_bkp AS
SELECT * FROM jobs;
SELECT * FROM jobs_bkp;





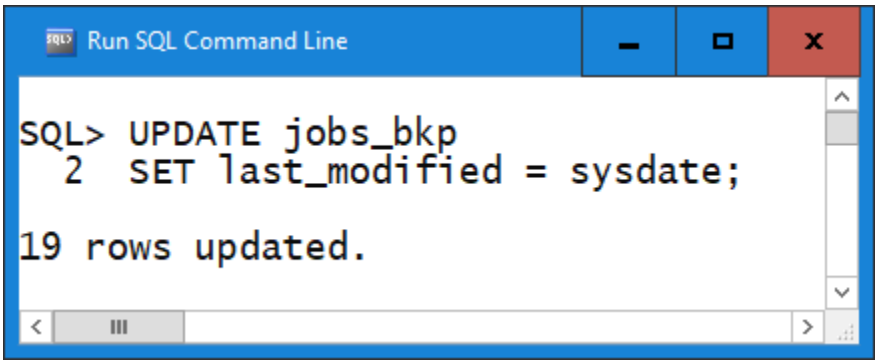
Part 2:

ALTER TABLE jobs_bkp
ADD last_modified DATE;
DESC jobs_bkp;



Part 3:

UPDATE jobs_bkp
SET last_modified = sysdate;
SELECT * FROM jobs_bkp;

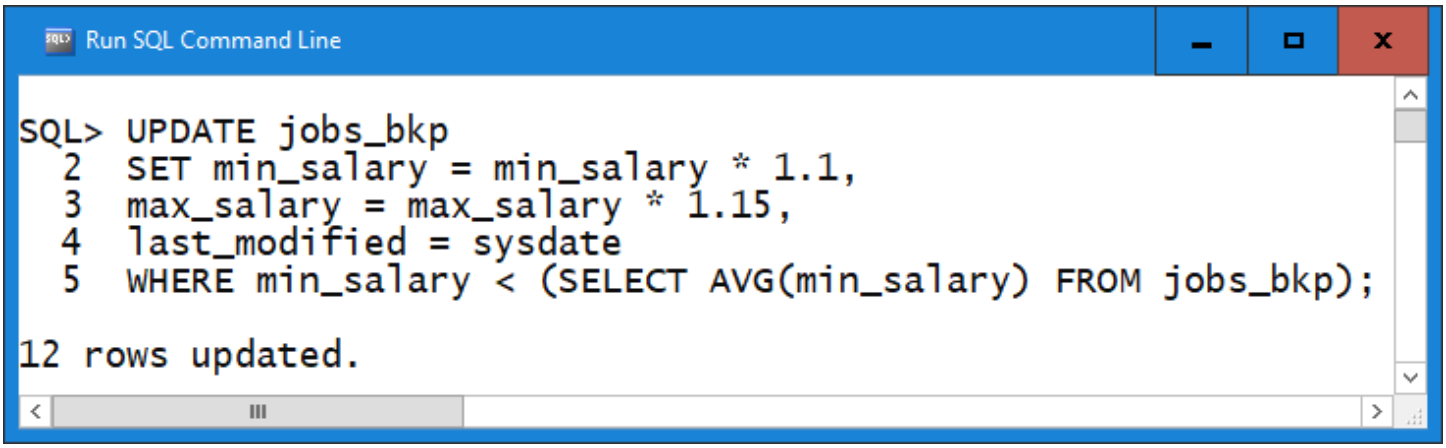


Run SQL Command Line

```
SQL> SELECT * FROM jobs_bkp;

JOB_ID     JOB_TITLE                           MIN_SALARY MAX_SALARY LAST_MODI
---------- ----------------------------------- ---------- ---------- ---------
AD_PRES    President                                20000      40000 06-AUG-20
AD_VP      Administration Vice President            15000      30000 06-AUG-20
AD_ASST    Administration Assistant                  3000       6000 06-AUG-20
FI_MGR     Finance Manager                           8200      16000 06-AUG-20
FI_ACCOUNT Accountant                                4200       9000 06-AUG-20
AC_MGR     Accounting Manager                        8200      16000 06-AUG-20
AC_ACCOUNT Public Accountant                         4200       9000 06-AUG-20
SA_MAN     Sales Manager                            10000      20000 06-AUG-20
SA_REP     Sales Representative                      6000      12000 06-AUG-20
PU_MAN     Purchasing Manager                        8000      15000 06-AUG-20
PU_CLERK   Purchasing Clerk                          2500       5500 06-AUG-20
ST_MAN     Stock Manager                             5500       8500 06-AUG-20
ST_CLERK   Stock Clerk                               2000       5000 06-AUG-20
SH_CLERK   Shipping Clerk                           2500       5500 06-AUG-20
IT_PROG    Programmer                                4000      10000 06-AUG-20
MK_MAN     Marketing Manager                         9000      15000 06-AUG-20
MK_REP     Marketing Representative                  4000       9000 06-AUG-20
HR_REP     Human Resources Representative            4000       9000 06-AUG-20
PR_REP     Public Relations Representative           4500      10500 06-AUG-20

19 rows selected.
```

Part 4

```
UPDATE jobs_bkp
SET min_salary = min_salary * 1.1,
max_salary = max_salary * 1.15,
last_modified = sysdate
WHERE min_salary < (SELECT AVG(min_salary) FROM jobs_bkp);
```




```
SELECT job_id, job_title, min_salary, max_salary,
TO_CHAR(last_modified, 'dd/mm/yyyy hh:mi:ss am') date_time_updated
FROM jobs_bkp;
```