```
/*
	Assignment 1: Basic SQL
	Statement: Write the following simple SQL Queries on the University Schema.
*/
/*
	Find the names of all the students whose total credits are greater than 100.
*/

select name
from student
where tot_cred > 100;

/*
	OUTPUT:-

	+--------+
	| name   |
	+--------+
	| Zhang  |
	| Chavez |
	| Tanaka |
	+--------+
	3 rows in set (0.00 sec)
*/

/*
	Find the course id and grades of all courses taken by any student named 'Tanaka'.
*/

select course_id, grade
from takes
where id = (select id
            from student
            where name = 'Tanaka');


/*
	OUTPUT:-
	+-----------+-------+
	| course_id | grade |
	+-----------+-------+
	| BIO-101   | A     |
	| BIO-301   | NULL  |
	+-----------+-------+
	2 rows in set (0.00 sec)

*/

/*
	Find the ID and name of instructors who have taught a course in the Comp. Sci. department, even
if they are themselves not from the       Comp. Sci. department. To test this query, make sure you add
appropriate data, and include the corresponding insert statements along             with your query.
*/
insert into instructor values ('10000', 'success', 'Elec. Eng.', '80000');
insert into teaches values ('10000', 'CS-101', '1', 'Fall', '2009');
select ID, name
from (      select ID
            from teaches natural join (   select course_id
                                          from course
                                          where dept_name = 'Comp. Sci.') as c) as x natural join
instructor
where dept_name != 'Comp. Sci.';

/*
	OUTPUT:-
	+-------+---------+
	| ID    | name    |
	+-------+---------+
	| 10000 | success |
	+-------+---------+
	1 row in set (0.00 sec)
*/
```

```
/*
	--Find the courses which are offered in both 'Fall' and 'Spring' semester (not necessarily in
the same year).
*/

select course_id, title
from (select s1.course_id
	from (select course_id, semester
		from section) as s1, (select course_id, semester
					from section) as s2
	where s1.course_id = s2.course_id and s1.semester = 'Fall' and s2.semester = 'Spring') as x
natural join course as c;

/*
	OUTPUT:-
	+-----------+--------------------------+
	| course_id | title                    |
	+-----------+--------------------------+
	| CS-101    | Intro. to Computer Science |
	+-----------+--------------------------+
	1 row in set (0.00 sec)
*/


/* OPTIONAL QUERIES */


/*
	Find the names of all the instructors from Comp. Sci. department.
*/

select name
from instructor
where dept_name = 'Comp. Sci.';

/*
	OUTPUT:-
	+------------+
	| name       |
	+------------+
	| Srinivasan |
	| Katz       |
	| Brandt     |
	+------------+
	3 rows in set (0.00 sec)
*/

/*
	Find the course id and titles of all courses taught by an instructor named 'Srinivasan'
*/

select it.course_id, title
from
	(select course_id
	from
		((select id
		from instructor
		where name = 'Srinivasan') as i natural join teaches)) as it natural join course
;

/*
	OUTPUT:-
	+-----------+--------------------------+
	| course_id | title                    |
	+-----------+--------------------------+
	| CS-101    | Intro. to Computer Science |
	| CS-315    | Robotics                 |
	| CS-347    | Database System Concepts |
	+-----------+--------------------------+
	3 rows in set (0.00 sec)

*/
```

```
/*
	Find names of instructors who have taught at least one course in Spring 2009.
*/

select distinct name
from
	(select Id
	 from teaches
	 where year = 2009 and semester = 'Spring') as t natural join instructor;

/*
	OUTPUT:-
	+--------+
	| name   |
	+--------+
	| Brandt |
	| Kim    |
	+--------+
	2 rows in set (0.01 sec)
*/


/*
	Statement :  Write the following Queries for Railway Schema.
*/

/*
	Find pairs of stations (station codes) that have a track (direct connection) with distance less
than 20Kms between them.
*/

select stcode1, stcode2
from track
where distance < 20
;

/*
	OUTPUT:-
	+---------+---------+
	| stcode1 | stcode2 |
	+---------+---------+
	| BYC     | DR      |
	| BYC     | KRL     |
	| CST     | BYC     |
	| CST     | DR      |
	| CST     | KRL     |
	| GRP     | TNA     |
	+---------+---------+
	6 rows in set (0.00 sec)
*/

/*
	Find the IDs of all the trains which have a stop at THANE
*/

select id
from
	  (select stcode
	   from station
	   where name = 'THANE') as s  natural join trainhalts;

/*
	OUTPUT:-
	+------+
	| id   |
	+------+
	| A65  |
	| KP11 |
	+------+
	2 rows in set (0.00 sec)
*/
```

```
/*
	Find the names of all trains that start at MUMBAI.
*/

select name
from
	(select id
	from
		(select stcode
		from station
		where name = 'MUMBAI') as s natural join trainhalts
	where seqno = 0) as x natural join train
;

/*
	OUTPUT:-
	+---------------+
	| name          |
	+---------------+
	| CST-AMR_LOCAL |
	| CST-KYN       |
	+---------------+
	2 rows in set (0.00 sec)
*/

/*
	List all the stations in order of visit by the train 'CST-AMR_LOCAL'.
*/

select name
from trainhalts natural join station
where id = (select id
		from train
		where name = 'CST-AMR_LOCAL')
order by seqno asc
;

/*
	OUTPUT:-
	+-----------+
	| name      |
	+-----------+
	| MUMBAI    |
	| BYCULLA   |
	| DADAR     |
	| KURLA     |
	| GHATKOPAR |
	| THANE     |
	| DOMBIVALI |
	| KALYAN    |
	| AMBARNATH |
	+-----------+
	9 rows in set (0.00 sec)
*/

/*
	Find the name of the trains which have stop at Thane, before the 6th station in the route of the
train.
*/


select name
from
	(select id
	 from trainhalts
	 where seqno < 6 and stcode = (select stcode
					from station
					where name = 'THANE')) as x natural join train
;
```

```
/*
      OUTPUT:-
      +---------------+
      | name          |
      +---------------+
      | CST-AMR_LOCAL |
      +---------------+
      1 row in set (0.00 sec)
*/
```