MySQL

Day 1

Create a sample database with the name "service_station".

```
mysql> create database service_station;
Query OK, 1 row affected (0.00 sec)

mysql> use service_station;
Database changed
```

Create models and write create table statements for the following entities and ensure these can be executed:

- Service Station - id, name, address, contact
- employees - id, name, age, contact
- customers - id, name, age, contact, emp_id (employee can also be a customer)
- vehicles - id, type(Car, Bike, Bus), brand, color, service_charge
- invoices - id, name_of_owner, vehicle, amount_total, employee_assigned

```
mysql> create table service_stations(id varchar(20) primary key, name varchar(20), address varchar(40), contact numeric(10));
Query OK, 0 rows affected (0.07 sec)

mysql> create table employees(id varchar(20), name varchar(20), age numeric(2),contact numeric(10));
Query OK, 0 rows affected (0.10 sec)

mysql> alter table employees add primary key(id);
Query OK, 0 rows affected (0.26 sec)
Records: 0 Duplicates: 0 Warnings: 0


mysql> create table customers(id varchar(20) primary key, name varchar(20),age numeric(2), contact numeric(10), emp_id varchar(20), FOREIGN KEY(emp_id) REFERENCES employees(id));
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> create table vehicles(id varchar(20) primary key, type enum('car','bike','bus'));
Query OK, 0 rows affected (0.08 sec)


mysql> create table invoices (id varchar(20) primary key, name_of_owner varchar(20), vehicle
varchar(20), amount_total numeric(6,2), employee_assigned varchar(20));
Query OK, 0 rows affected (0.31 sec)


mysql> ALTER TABLE invoices add constraint fk1 foreign key(employee_assigned) references
employees(id);
Query OK, 0 rows affected (0.20 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Drop tables and database.

```
mysql> drop table vehicles;
Query OK, 0 rows affected (0.50 sec)

mysql> drop table invoices;;
Query OK, 0 rows affected (0.08 sec)


mysql> drop table customers;;
Query OK, 0 rows affected (0.06 sec)


mysql> drop table service_station;
Query OK, 0 rows affected (0.08 sec)

mysql> drop table employees;
Query OK, 0 rows affected (0.08 sec)

mysql> drop database service_station;
Query OK, 0 rows affected (0.00 sec)
```

Day 2 and 3

```
DROP TABLE IF EXISTS `students`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `students` (
`id` bigint(19) NOT NULL AUTO_INCREMENT,
`name` varchar(100) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=100007 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `students`
--

LOCK TABLES `students` WRITE;
/*!40000 ALTER TABLE `students` DISABLE KEYS */;
INSERT INTO `students` VALUES
(100001,'Thor'),(100002,'Hulk'),(100003,'Daredevil'),(100004,'X
Man'),(100005,'Ironman'),(100006,'Quicksilver');
/*!40000 ALTER TABLE `students` ENABLE KEYS */;
UNLOCK TABLES;


DROP TABLE IF EXISTS `marks`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `marks` (
`id` bigint(19) NOT NULL AUTO_INCREMENT,
`student_id` bigint(19) NOT NULL,
`subject_id` bigint(19) NOT NULL,
`quarterly` int(11) DEFAULT NULL,
`half_yearly` int(11) DEFAULT NULL,
```

```
`annual` int(11) DEFAULT NULL,
`year` int(11) DEFAULT NULL,
`grade` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_marks_student_id` (`student_id`),
CONSTRAINT `fk_marks_student_id` FOREIGN KEY (`student_id`) REFERENCES
`students` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1126 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `marks`
--

LOCK TABLES `marks` WRITE;
/*!40000 ALTER TABLE `marks` DISABLE KEYS */;
INSERT INTO `marks` VALUES
(1001,100001,1,NULL,79,91,2003,6),(1002,100002,1,35,49,77,2003,6),(1003,100003,1,
100,97,95,2003,6),(1004,100004,1,38,38,65,2003,6),(1005,100005,1,73,40,100,2003,6)
,(1006,100001,2,NULL,30,76,2003,6),(1007,100002,2,37,45,87,2003,6),(1008,100003,2
,93,91,98,2003,6),(1009,100004,2,93,59,63,2003,6),(1010,100005,2,34,89,45,2003,6),(
1011,100001,3,NULL,86,89,2003,6),(1012,100002,3,46,76,48,2003,6),(1013,100003,3,
46,NULL,83,2003,6),(1014,100004,3,71,74,31,2003,6),(1015,100005,3,34,54,36,2003,6
),(1016,100001,4,NULL,NULL,80,2003,6),(1017,100002,4,52,43,31,2003,6),(1018,1000
03,4,91,95,99,2003,6),(1019,100004,4,46,76,39,2003,6),(1020,100005,4,80,41,94,2003
,6),(1021,100001,5,NULL,31,88,2003,6),(1022,100002,5,33,44,53,2003,6),(1023,10000
3,5,98,92,90,2003,6),(1024,100004,5,52,63,63,2003,6),(1025,100005,5,56,60,48,2003,
6),(1026,100001,1,59,34,57,2004,7),(1027,100002,1,47,37,94,2004,7),(1028,100003,1,
47,80,97,2004,7),(1029,100004,1,89,43,68,2004,7),(1030,100005,1,72,82,47,2004,7),(1
031,100001,2,44,54,31,2004,7),(1032,100002,2,85,96,89,2004,7),(1033,100003,2,84,6
3,57,2004,7),(1034,100004,2,83,97,53,2004,7),(1035,100005,2,53,30,80,2004,7),(1036,
100001,3,44,90,54,2004,7),(1037,100002,3,53,36,87,2004,7),(1038,100003,3,64,55,39,
2004,7),(1039,100004,3,95,36,54,2004,7),(1040,100005,3,66,87,37,2004,7),(1041,1000
01,4,31,65,69,2004,7),(1042,100002,4,98,30,94,2004,7),(1043,100003,4,43,60,83,2004
,7),(1044,100004,4,76,79,40,2004,7),(1045,100005,4,66,43,75,2004,7),(1046,100001,5,
58,51,98,2004,7),(1047,100002,5,41,92,99,2004,7),(1048,100003,5,86,56,43,2004,7),(1
049,100004,5,35,90,92,2004,7),(1050,100005,5,36,35,48,2004,7),(1051,100001,1,96,5
7,94,2005,8),(1052,100002,1,68,41,50,2005,8),(1053,100003,1,60,83,73,2005,8),(1054,
100004,1,53,81,97,2005,8),(1055,100005,1,64,81,39,2005,8),(1056,100001,2,77,82,36,
2005,8),(1057,100002,2,73,57,52,2005,8),(1058,100003,2,38,35,81,2005,8),(1059,1000
04,2,69,46,44,2005,8),(1060,100005,2,35,95,39,2005,8),(1061,100001,3,32,70,58,2005
,8),(1062,100002,3,43,48,65,2005,8),(1063,100003,3,81,38,90,2005,8),(1064,100004,3,
88,90,92,2005,8),(1065,100005,3,34,95,76,2005,8),(1066,100001,4,64,69,87,2005,8),(1
067,100002,4,62,38,95,2005,8),(1068,100003,4,79,49,86,2005,8),(1069,100004,4,68,3
3,33,2005,8),(1070,100005,4,72,39,84,2005,8),(1071,100001,5,65,77,100,2005,8),(107
```

2,100002,5,82,90,33,2005,8),(1073,100003,5,76,44,55,2005,8),(1074,100004,5,96,76,46,2005,8),(1075,100005,5,50,75,49,2005,8),(1076,100001,1,67,84,51,2006,9),(1077,100002,1,63,68,46,2006,9),(1078,100003,1,89,96,31,2006,9),(1079,100004,1,86,74,82,2006,9),(1080,100005,1,91,49,70,2006,9),(1081,100001,2,65,90,67,2006,9),(1082,100002,2,77,38,75,2006,9),(1083,100003,2,100,99,95,2006,9),(1084,100004,2,100,37,40,2006,9),(1085,100005,2,85,69,37,2006,9),(1086,100001,3,NULL,NULL,32,2006,9),(1087,100002,3,98,34,31,2006,9),(1088,100003,3,65,83,36,2006,9),(1089,100004,3,82,75,67,2006,9),(1090,100005,3,93,78,39,2006,9),(1091,100001,4,83,75,59,2006,9),(1092,100002,4,81,56,30,2006,9),(1093,100003,4,94,60,68,2006,9),(1094,100004,4,59,92,47,2006,9),(1095,100005,4,76,82,83,2006,9),(1096,100001,5,73,70,87,2006,9),(1097,100002,5,50,52,36,2006,9),(1098,100003,5,57,96,88,2006,9),(1099,100004,5,54,77,51,2006,9),(1100,100005,5,86,66,91,2006,9),(1101,100001,1,NULL,NULL,NULL,2007,10),(1102,100002,1,93,31,79,2007,10),(1103,100003,1,81,92,69,2007,10),(1104,100004,1,32,31,76,2007,10),(1105,100005,1,82,65,87,2007,10),(1106,100001,2,62,80,36,2007,10),(1107,100002,2,34,49,87,2007,10),(1108,100003,2,95,89,89,2007,10),(1109,100004,2,50,46,76,2007,10),(1110,100005,2,86,87,84,2007,10),(1111,100001,3,91,64,80,2007,10),(1112,100002,3,52,60,94,2007,10),(1113,100003,3,50,81,58,2007,10),(1114,100004,3,65,39,46,2007,10),(1115,100005,3,44,79,91,2007,10),(1116,100001,4,55,96,55,2007,10),(1117,100002,4,82,67,94,2007,10),(1118,100003,4,85,76,47,2007,10),(1119,100004,4,79,89,89,2007,10),(1120,100005,4,72,56,57,2007,10),(1121,100001,5,93,52,41,2007,10),(1122,100002,5,42,63,75,2007,10),(1123,100003,5,88,31,43,2007,10),(1124,100004,5,57,36,67,2007,10),(1125,100005,5,87,99,93,2007,10);
/*!40000 ALTER TABLE `marks` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `medals`
--

DROP TABLE IF EXISTS `medals`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `medals` (
`id` bigint(19) NOT NULL AUTO_INCREMENT,
`student_id` bigint(19) NOT NULL,
`game_id` bigint(19) NOT NULL,
`medal_won` varchar(10) DEFAULT NULL,
`year` int(11) DEFAULT NULL,
`grade` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_medals_student_id` (`student_id`),
CONSTRAINT `fk_medals_student_id` FOREIGN KEY (`student_id`) REFERENCES
`students` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=114 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

```
--
-- Dumping data for table `medals`
--

LOCK TABLES `medals` WRITE;
/*!40000 ALTER TABLE `medals` DISABLE KEYS */;
INSERT INTO `medals` VALUES
(101,100003,5,'gold',2003,6),(102,100001,5,'silver',2003,6),(103,100002,4,'silver',2003,
6),(104,100003,1,'gold',2003,6),(105,100003,3,'bronze',2004,7),(106,100003,4,'silver',2
004,7),(107,100002,2,'silver',2004,7),(108,100002,4,'bronze',2004,7),(109,100002,5,'gol
d',2004,7),(110,100003,3,'bronze',2005,8),(111,100001,2,'gold',2005,8),(112,100001,3,'
bronze',2005,8),(113,100001,4,'bronze',2005,8);
/*!40000 ALTER TABLE `medals` ENABLE KEYS */;
UNLOCK TABLES;
```

Queries that uses “students” table:

1. Select all the students

```
mysql> select * from students;
+--------+-------------+
| id | name |
+--------+-------------+
| 100001 | Thor |
| 100002 | Hulk |
| 100003 | Daredevil |
| 100004 | X Man |
| 100005 | Ironman |
| 100006 | Quicksilver |
+--------+-------------+
6 rows in set (0.00 sec)
```

2. Select all the students whose names starts with “H”

```
mysql> select * from students where name like 'H%';
+--------+------+
```

```
| id | name |
+--------+------+
| 100002 | Hulk |
+--------+------+
1 row in set (0.00 sec)
```

3. Select all the students whose name has the alphabet “a”

```
mysql> select * from students where name like '%a%';
+--------+-----------+
| id | name |
+--------+-----------+
| 100003 | Daredevil |
| 100004 | X Man |
| 100005 | Ironman |
+--------+-----------+
3 rows in set (0.00 sec)
```

3. Select all the students and list the results sorted in alphabetical order(a-z).

```
mysql> select * from students order by name;
+--------+-------------+
| id | name |
+--------+-------------+
| 100003 | Daredevil |
| 100002 | Hulk |
| 100005 | Ironman |
| 100006 | Quicksilver |
| 100001 | Thor |
| 100004 | X Man |
+--------+-------------+
6 rows in set (0.00 sec)
```

4. List the first “2” students with the results sorted in the alphabetical order(a-z).

```
mysql> select * from students order by name limit 2;
+--------+-----------+
| id | name |
+--------+-----------+
| 100003 | Daredevil |
```

```
| 100002 | Hulk |
+--------+-----------+
2 rows in set (0.00 sec)
```

5. List the next “2” students(3rd and 4th) when they are sorted in the alphabetical order.

```
mysql> select * from students order by name limit 2,2;
+--------+-------------+
| id | name |
+--------+-------------+
| 100005 | Ironman |
| 100006 | Quicksilver |
+--------+-------------+
2 rows in set (0.00 sec)
```

Queries that uses “marks” table:

1. Select the students who has not appeared in the annual exams. Format: All columns of the “marks” table.

```
mysql> select * from marks where annual is NULL;
+------+------------+------------+-----------+-------------+--------+------+-------+
| id | student_id | subject_id | quarterly | half_yearly | annual | year | grade |
+------+------------+------------+-----------+-------------+--------+------+-------+
| 1101 | 100001 | 1 | NULL | NULL | NULL | 2007 | 10 |
+------+------------+------------+-----------+-------------+--------+------+-------+
1 row in set (0.00 sec)
```

2. Select the students who has not appeared in the annual exams during the year “2005”.
Format: student_id, subject_id, year

mysql> select student_id,subject_id, year from marks where annual is NULL and year='2005';
Empty set (0.00 sec)

3. Select the students who has appeared in one of the exams - quarterly, half_yearly or annual.
Format: student_id, subject_id, year

mysql> select student_id,subject_id, year from marks where quarterly is not NULL or half_yearly is not NULL or year is not NULL;

student_id	subject_id	year
100001	1	2003
100002	1	2003
100003	1	2003
100004	1	2003
100005	1	2003
100001	2	2003
100002	2	2003
100003	2	2003
100004	2	2003
100005	2	2003
100001	3	2003
100002	3	2003
100003	3	2003
100004	3	2003
100005	3	2003
100001	4	2003
100002	4	2003
100003	4	2003
100004	4	2003
100005	4	2003
100001	5	2003
100002	5	2003
100003	5	2003
100004	5	2003
100005	5	2003
100001	1	2004

100002	1	2004
100003	1	2004
100004	1	2004
100005	1	2004
100001	2	2004
100002	2	2004
100003	2	2004
100004	2	2004
100005	2	2004
100001	3	2004
100002	3	2004
100003	3	2004
100004	3	2004
100005	3	2004
100001	4	2004
100002	4	2004
100003	4	2004
100004	4	2004
100005	4	2004
100001	5	2004
100002	5	2004
100003	5	2004
100004	5	2004
100005	5	2004
100001	1	2005
100002	1	2005
100003	1	2005
100004	1	2005
100005	1	2005
100001	2	2005
100002	2	2005
100003	2	2005
100004	2	2005
100005	2	2005
100001	3	2005
100002	3	2005
100003	3	2005
100004	3	2005
100005	3	2005
100001	4	2005
100002	4	2005
100003	4	2005
100004	4	2005
100005	4	2005
100001	5	2005

100002	5	2005
100003	5	2005
100004	5	2005
100005	5	2005
100001	1	2006
100002	1	2006
100003	1	2006
100004	1	2006
100005	1	2006
100001	2	2006
100002	2	2006
100003	2	2006
100004	2	2006
100005	2	2006
100001	3	2006
100002	3	2006
100003	3	2006
100004	3	2006
100005	3	2006
100001	4	2006
100002	4	2006
100003	4	2006
100004	4	2006
100005	4	2006
100001	5	2006
100002	5	2006
100003	5	2006
100004	5	2006
100005	5	2006
100001	1	2007
100002	1	2007
100003	1	2007
100004	1	2007
100005	1	2007
100001	2	2007
100002	2	2007
100003	2	2007
100004	2	2007
100005	2	2007
100001	3	2007
100002	3	2007
100003	3	2007
100004	3	2007
100005	3	2007
100001	4	2007

```
| 100002 | 4 | 2007 |
| 100003 | 4 | 2007 |
| 100004 | 4 | 2007 |
| 100005 | 4 | 2007 |
| 100001 | 5 | 2007 |
| 100002 | 5 | 2007 |
| 100003 | 5 | 2007 |
| 100004 | 5 | 2007 |
| 100005 | 5 | 2007 |
+------------+------------+------+
125 rows in set (0.00 sec)
```

4. Select the students who has scored more than 90 in all the exams - quarterly, half_yearly and annual.
Format: student_id, subject_id, year, quarterly, half_yearly, annual

```
mysql> select student_id,subject_id, year,quarterly,half_yearly,annual from marks
where quarterly >90 and half_yearly>90 and annual>90;
+------------+------------+------+-----------+-------------+--------+
| student_id | subject_id | year | quarterly | half_yearly | annual |
+------------+------------+------+-----------+-------------+--------+
| 100003 | 1 | 2003 | 100 | 97 | 95 |
| 100003 | 2 | 2003 | 93 | 91 | 98 |
| 100003 | 4 | 2003 | 91 | 95 | 99 |
| 100003 | 2 | 2006 | 100 | 99 | 95 |
+------------+------------+------+-----------+-------------+--------+
4 rows in set (0.00 sec)
```

5. List the average marks(in quarterly, half_yearly & annual) for each subject scored for the year.
Format: student_id, subject_id, average, year

```
mysql> select student_id,subject_id,sum(quarterly+half_yearly+annual)/3 average,year
from marks group by student_id,subject_id,year;
+------------+------------+---------+------+
| student_id | subject_id | average | year |
+------------+------------+---------+------+
| 100001 | 1 | NULL | 2003 |
| 100001 | 1 | 50.0000 | 2004 |
| 100001 | 1 | 82.3333 | 2005 |
| 100001 | 1 | 67.3333 | 2006 |
| 100001 | 1 | NULL | 2007 |
```

100001	2	NULL	2003
100001	2	43.0000	2004
100001	2	65.0000	2005
100001	2	74.0000	2006
100001	2	59.3333	2007
100001	3	NULL	2003
100001	3	62.6667	2004
100001	3	53.3333	2005
100001	3	NULL	2006
100001	3	78.3333	2007
100001	4	NULL	2003
100001	4	55.0000	2004
100001	4	73.3333	2005
100001	4	72.3333	2006
100001	4	68.6667	2007
100001	5	NULL	2003
100001	5	69.0000	2004
100001	5	80.6667	2005
100001	5	76.6667	2006
100001	5	62.0000	2007
100002	1	53.6667	2003
100002	1	59.3333	2004
100002	1	53.0000	2005
100002	1	59.0000	2006
100002	1	67.6667	2007
100002	2	56.3333	2003
100002	2	90.0000	2004
100002	2	60.6667	2005
100002	2	63.3333	2006
100002	2	56.6667	2007
100002	3	56.6667	2003
100002	3	58.6667	2004
100002	3	52.0000	2005
100002	3	54.3333	2006
100002	3	68.6667	2007
100002	4	42.0000	2003
100002	4	74.0000	2004
100002	4	65.0000	2005
100002	4	55.6667	2006
100002	4	81.0000	2007
100002	5	43.3333	2003
100002	5	77.3333	2004
100002	5	68.3333	2005

| 100002 | 5 | 60.0000 | 2007 |

100003	1	97.3333	2003
100003	1	74.6667	2004
100003	1	72.0000	2005
100003	1	72.0000	2006
100003	1	80.6667	2007
100003	2	94.0000	2003
100003	2	68.0000	2004
100003	2	51.3333	2005
100003	2	98.0000	2006
100003	2	91.0000	2007
100003	3	NULL	2003
100003	3	52.6667	2004
100003	3	69.6667	2005
100003	3	61.3333	2006
100003	3	63.0000	2007
100003	4	95.0000	2003
100003	4	62.0000	2004
100003	4	71.3333	2005
100003	4	74.0000	2006
100003	4	69.3333	2007
100003	5	93.3333	2003
100003	5	61.6667	2004
100003	5	58.3333	2005
100003	5	80.3333	2006
100003	5	54.0000	2007
100004	1	47.0000	2003
100004	1	66.6667	2004
100004	1	77.0000	2005
100004	1	80.6667	2006
100004	1	46.3333	2007
100004	2	71.6667	2003
100004	2	77.6667	2004
100004	2	53.0000	2005
100004	2	59.0000	2006
100004	2	57.3333	2007
100004	3	58.6667	2003
100004	3	61.6667	2004
100004	3	90.0000	2005
100004	3	74.6667	2006
100004	3	50.0000	2007
100004	4	53.6667	2003
100004	4	65.0000	2004
100004	4	44.6667	2005
100004	4	66.0000	2006
100004	4	85.6667	2007

100004	5	59.3333	2003
100004	5	72.3333	2004
100004	5	72.6667	2005
100004	5	60.6667	2006
100004	5	53.3333	2007
100005	1	71.0000	2003
100005	1	67.0000	2004
100005	1	61.3333	2005
100005	1	70.0000	2006
100005	1	78.0000	2007
100005	2	56.0000	2003
100005	2	54.3333	2004
100005	2	56.3333	2005
100005	2	63.6667	2006
100005	2	85.6667	2007
100005	3	41.3333	2003
100005	3	63.3333	2004
100005	3	68.3333	2005
100005	3	70.0000	2006
100005	3	71.3333	2007
100005	4	71.6667	2003
100005	4	61.3333	2004
100005	4	65.0000	2005
100005	4	80.3333	2006
100005	4	61.6667	2007
100005	5	54.6667	2003
100005	5	39.6667	2004
100005	5	58.0000	2005
100005	5	81.0000	2006
100005	5	93.0000	2007
+------------+------------+---------+------+
125 rows in set (0.00 sec)

6. List the average marks(in quarterly, half_yearly & annual) for each subject scored for the years 2003 & 2004
Format: student_id, subject_id, average, year

mysql> select student_id,subject_id,sum(quarterly+half_yearly+annual)/3 average,year from marks where year in('2003','2004') group by student_id,subject_id,year;
+------------+------------+---------+------+
| student_id | subject_id | average | year |
+------------+------------+---------+------+

100001	1	NULL	2003
100001	1	50.0000	2004
100001	2	NULL	2003
100001	2	43.0000	2004
100001	3	NULL	2003
100001	3	62.6667	2004
100001	4	NULL	2003
100001	4	55.0000	2004
100001	5	NULL	2003
100001	5	69.0000	2004
100002	1	53.6667	2003
100002	1	59.3333	2004
100002	2	56.3333	2003
100002	2	90.0000	2004
100002	3	56.6667	2003
100002	3	58.6667	2004
100002	4	42.0000	2003
100002	4	74.0000	2004
100002	5	43.3333	2003
100002	5	77.3333	2004
100003	1	97.3333	2003
100003	1	74.6667	2004
100003	2	94.0000	2003
100003	2	68.0000	2004
100003	3	NULL	2003
100003	3	52.6667	2004
100003	4	95.0000	2003
100003	4	62.0000	2004
100003	5	93.3333	2003
100003	5	61.6667	2004
100004	1	47.0000	2003
100004	1	66.6667	2004
100004	2	71.6667	2003
100004	2	77.6667	2004
100004	3	58.6667	2003
100004	3	61.6667	2004
100004	4	53.6667	2003
100004	4	65.0000	2004
100004	5	59.3333	2003
100004	5	72.3333	2004
100005	1	71.0000	2003
100005	1	67.0000	2004
100005	2	56.0000	2003
100005	2	54.3333	2004
100005	3	41.3333	2003

```
| 100005 | 3 | 63.3333 | 2004 |
| 100005 | 4 | 71.6667 | 2003 |
| 100005 | 4 | 61.3333 | 2004 |
| 100005 | 5 | 54.6667 | 2003 |
| 100005 | 5 | 39.6667 | 2004 |
+------------+------------+---------+------+
50 rows in set (0.00 sec)
```

Joins & SubQueries

Write the queries from the "marks" table using inner join with the table "students" table to replace student_id with name.

1. Select the students who has not appeared in the annual exams. Format: All columns of the “marks” table.

```
mysql> select m.id,name ,subject_id,quarterly,half_yearly,annual,year,grade from marks
m,students s where m.student_id=s.id and annual is null;
+------+------+------------+-----------+-------------+--------+------+-------+
| id | name | subject_id | quarterly | half_yearly | annual | year | grade |
+------+------+------------+-----------+-------------+--------+------+-------+
| 1101 | Thor | 1 | NULL | NULL | NULL | 2007 | 10 |
+------+------+------------+-----------+-------------+--------+------+-------+
1 row in set (0.00 sec)
```

2. Select the students who has not appeared in the annual exams during the year “2005”. Format: student_id, subject_id, year

```
mysql> select name ,subject_id,year from marks m,students s where m.student_id=s.id
and year='2005' and annual is null;
Empty set (0.00 sec)
```

3. Select the students who has appeared in one of the exams - quarterly, half_yearly or annual. Format: student_id, subject_id, year

```
mysql> select name ,subject_id,year from marks m,students s where m.student_id=s.id
and (quarterly is not null or half_yearly is not null or annual is not null);
+-----------+------------+------+
| name | subject_id | year |
+-----------+------------+------+
| Thor | 1 | 2003 |
| Thor | 2 | 2003 |
| Thor | 3 | 2003 |
| Thor | 4 | 2003 |
| Thor | 5 | 2003 |
| Thor | 1 | 2004 |
| Thor | 2 | 2004 |
| Thor | 3 | 2004 |
| Thor | 4 | 2004 |
| Thor | 5 | 2004 |
| Thor | 1 | 2005 |
| Thor | 2 | 2005 |
| Thor | 3 | 2005 |
| Thor | 4 | 2005 |
| Thor | 5 | 2005 |
| Thor | 1 | 2006 |
| Thor | 2 | 2006 |
| Thor | 3 | 2006 |
| Thor | 4 | 2006 |
| Thor | 5 | 2006 |
| Thor | 2 | 2007 |
| Thor | 3 | 2007 |
| Thor | 4 | 2007 |
| Thor | 5 | 2007 |
| Hulk | 1 | 2003 |
| Hulk | 2 | 2003 |
| Hulk | 3 | 2003 |
| Hulk | 4 | 2003 |
| Hulk | 5 | 2003 |
| Hulk | 1 | 2004 |
| Hulk | 2 | 2004 |
| Hulk | 3 | 2004 |
| Hulk | 4 | 2004 |
| Hulk | 5 | 2004 |
| Hulk | 1 | 2005 |
| Hulk | 2 | 2005 |
| Hulk | 3 | 2005 |
| Hulk | 4 | 2005 |
| Hulk | 5 | 2005 |
```

Hulk	1	2006
Hulk	2	2006
Hulk	3	2006
Hulk	4	2006
Hulk	5	2006
Hulk	1	2007
Hulk	2	2007
Hulk	3	2007
Hulk	4	2007
Hulk	5	2007
Daredevil	1	2003
Daredevil	2	2003
Daredevil	3	2003
Daredevil	4	2003
Daredevil	5	2003
Daredevil	1	2004
Daredevil	2	2004
Daredevil	3	2004
Daredevil	4	2004
Daredevil	5	2004
Daredevil	1	2005
Daredevil	2	2005
Daredevil	3	2005
Daredevil	4	2005
Daredevil	5	2005
Daredevil	1	2006
Daredevil	2	2006
Daredevil	3	2006
Daredevil	4	2006
Daredevil	5	2006
Daredevil	1	2007
Daredevil	2	2007
Daredevil	3	2007
Daredevil	4	2007
Daredevil	5	2007
X Man	1	2003
X Man	2	2003
X Man	3	2003
X Man	4	2003
X Man	5	2003
X Man	1	2004
X Man	2	2004
X Man	3	2004
X Man	4	2004
X Man	5	2004

X Man	1	2005
X Man	2	2005
X Man	3	2005
X Man	4	2005
X Man	5	2005
X Man	1	2006
X Man	2	2006
X Man	3	2006
X Man	4	2006
X Man	5	2006
X Man	1	2007
X Man	2	2007
X Man	3	2007
X Man	4	2007
X Man	5	2007
Ironman	1	2003
Ironman	2	2003
Ironman	3	2003
Ironman	4	2003
Ironman	5	2003
Ironman	1	2004
Ironman	2	2004
Ironman	3	2004
Ironman	4	2004
Ironman	5	2004
Ironman	1	2005
Ironman	2	2005
Ironman	3	2005
Ironman	4	2005
Ironman	5	2005
Ironman	1	2006
Ironman	2	2006
Ironman	3	2006
Ironman	4	2006
Ironman	5	2006
Ironman	1	2007
Ironman	2	2007
Ironman	3	2007
Ironman	4	2007
Ironman	5	2007
+-----------+------------+------+
124 rows in set (0.00 sec)

4. Select the students who has scored more than 90 in all the exams - quarterly, half_yearly and annual.
Format: student_id, subject_id, year, quarterly, half_yearly, annual

```
mysql> select name ,subject_id,year,quarterly,half_yearly,annual from marks
m,students s where m.student_id=s.id and quarterly>90 and half_yearly>90 and
annual>90;
+-----------+------------+------+-----------+------------+--------+
| name | subject_id | year | quarterly | half_yearly | annual |
+-----------+------------+------+-----------+------------+--------+
| Daredevil | 1 | 2003 | 100 | 97 | 95 |
| Daredevil | 2 | 2003 | 93 | 91 | 98 |
| Daredevil | 4 | 2003 | 91 | 95 | 99 |
| Daredevil | 2 | 2006 | 100 | 99 | 95 |
+-----------+------------+------+-----------+------------+--------+
4 rows in set (0.00 sec)
```

5. List the average marks(in quarterly, half_yearly & annual) for each subject scored for the year.
Format: student_id, subject_id, average, year

```
mysql> select name ,subject_id,sum(quarterly+half_yearly+annual)/3 average,year from
marks m,students s where m.student_id=s.id group by student_id,subject_id,year;
+-----------+------------+---------+------+
| name | subject_id | average | year |
+-----------+------------+---------+------+
| Thor | 1 | NULL | 2003 |
| Thor | 1 | 50.0000 | 2004 |
| Thor | 1 | 82.3333 | 2005 |
| Thor | 1 | 67.3333 | 2006 |
| Thor | 1 | NULL | 2007 |
| Thor | 2 | NULL | 2003 |
| Thor | 2 | 43.0000 | 2004 |
| Thor | 2 | 65.0000 | 2005 |
| Thor | 2 | 74.0000 | 2006 |
| Thor | 2 | 59.3333 | 2007 |
| Thor | 3 | NULL | 2003 |
| Thor | 3 | 62.6667 | 2004 |
| Thor | 3 | 53.3333 | 2005 |
| Thor | 3 | NULL | 2006 |
| Thor | 3 | 78.3333 | 2007 |
```

Thor	4	NULL	2003
Thor	4	55.0000	2004
Thor	4	73.3333	2005
Thor	4	72.3333	2006
Thor	4	68.6667	2007
Thor	5	NULL	2003
Thor	5	69.0000	2004
Thor	5	80.6667	2005
Thor	5	76.6667	2006
Thor	5	62.0000	2007
Hulk	1	53.6667	2003
Hulk	1	59.3333	2004
Hulk	1	53.0000	2005
Hulk	1	59.0000	2006
Hulk	1	67.6667	2007
Hulk	2	56.3333	2003
Hulk	2	90.0000	2004
Hulk	2	60.6667	2005
Hulk	2	63.3333	2006
Hulk	2	56.6667	2007
Hulk	3	56.6667	2003
Hulk	3	58.6667	2004
Hulk	3	52.0000	2005
Hulk	3	54.3333	2006
Hulk	3	68.6667	2007
Hulk	4	42.0000	2003
Hulk	4	74.0000	2004
Hulk	4	65.0000	2005
Hulk	4	55.6667	2006
Hulk	4	81.0000	2007
Hulk	5	43.3333	2003
Hulk	5	77.3333	2004
Hulk	5	68.3333	2005
Hulk	5	46.0000	2006
Hulk	5	60.0000	2007
Daredevil	1	97.3333	2003
Daredevil	1	74.6667	2004
Daredevil	1	72.0000	2005
Daredevil	1	72.0000	2006
Daredevil	1	80.6667	2007
Daredevil	2	94.0000	2003
Daredevil	2	68.0000	2004
Daredevil	2	51.3333	2005
Daredevil	2	98.0000	2006
Daredevil	2	91.0000	2007

Daredevil	3	NULL	2003
Daredevil	3	52.6667	2004
Daredevil	3	69.6667	2005
Daredevil	3	61.3333	2006
Daredevil	3	63.0000	2007
Daredevil	4	95.0000	2003
Daredevil	4	62.0000	2004
Daredevil	4	71.3333	2005
Daredevil	4	74.0000	2006
Daredevil	4	69.3333	2007
Daredevil	5	93.3333	2003
Daredevil	5	61.6667	2004
Daredevil	5	58.3333	2005
Daredevil	5	80.3333	2006
Daredevil	5	54.0000	2007
X Man	1	47.0000	2003
X Man	1	66.6667	2004
X Man	1	77.0000	2005
X Man	1	80.6667	2006
X Man	1	46.3333	2007
X Man	2	71.6667	2003
X Man	2	77.6667	2004
X Man	2	53.0000	2005
X Man	2	59.0000	2006
X Man	2	57.3333	2007
X Man	3	58.6667	2003
X Man	3	61.6667	2004
X Man	3	90.0000	2005
X Man	3	74.6667	2006
X Man	3	50.0000	2007
X Man	4	53.6667	2003
X Man	4	65.0000	2004
X Man	4	44.6667	2005
X Man	4	66.0000	2006
X Man	4	85.6667	2007
X Man	5	59.3333	2003
X Man	5	72.3333	2004
X Man	5	72.6667	2005
X Man	5	60.6667	2006
X Man	5	53.3333	2007
Ironman	1	71.0000	2003
Ironman	1	67.0000	2004
Ironman	1	61.3333	2005
Ironman	1	70.0000	2006
Ironman	1	78.0000	2007

Ironman	2	56.0000	2003
Ironman	2	54.3333	2004
Ironman	2	56.3333	2005
Ironman	2	63.6667	2006
Ironman	2	85.6667	2007
Ironman	3	41.3333	2003
Ironman	3	63.3333	2004
Ironman	3	68.3333	2005
Ironman	3	70.0000	2006
Ironman	3	71.3333	2007
Ironman	4	71.6667	2003
Ironman	4	61.3333	2004
Ironman	4	65.0000	2005
Ironman	4	80.3333	2006
Ironman	4	61.6667	2007
Ironman	5	54.6667	2003
Ironman	5	39.6667	2004
Ironman	5	58.0000	2005
Ironman	5	81.0000	2006
Ironman	5	93.0000	2007
+-----------+------------+---------+------+
125 rows in set (0.00 sec)

6. List the average marks(in quarterly, half_yearly & annual) for each subject scored for the years 2003 & 2004
Format: student_id, subject_id, average, year

mysql> select name ,subject_id,sum(quarterly+half_yearly+annual)/3 average,year from marks m,students s where m.student_id=s.id and year in('2003','2004') group by student_id,subject_id,year;
+-----------+------------+---------+------+
| name | subject_id | average | year |
+-----------+------------+---------+------+
Thor	1	NULL	2003
Thor	1	50.0000	2004
Thor	2	NULL	2003
Thor	2	43.0000	2004
Thor	3	NULL	2003
Thor	3	62.6667	2004
Thor	4	NULL	2003
Thor	4	55.0000	2004
Thor	5	NULL	2003
Thor	5	69.0000	2004

Hulk	1	53.6667	2003
Hulk	1	59.3333	2004
Hulk	2	56.3333	2003
Hulk	2	90.0000	2004
Hulk	3	56.6667	2003
Hulk	3	58.6667	2004
Hulk	4	42.0000	2003
Hulk	4	74.0000	2004
Hulk	5	43.3333	2003
Hulk	5	77.3333	2004
Daredevil	1	97.3333	2003
Daredevil	1	74.6667	2004
Daredevil	2	94.0000	2003
Daredevil	2	68.0000	2004
Daredevil	3	NULL	2003
Daredevil	3	52.6667	2004
Daredevil	4	95.0000	2003
Daredevil	4	62.0000	2004
Daredevil	5	93.3333	2003
Daredevil	5	61.6667	2004
X Man	1	47.0000	2003
X Man	1	66.6667	2004
X Man	2	71.6667	2003
X Man	2	77.6667	2004
X Man	3	58.6667	2003
X Man	3	61.6667	2004
X Man	4	53.6667	2003
X Man	4	65.0000	2004
X Man	5	59.3333	2003
X Man	5	72.3333	2004
Ironman	1	71.0000	2003
Ironman	1	67.0000	2004
Ironman	2	56.0000	2003
Ironman	2	54.3333	2004
Ironman	3	41.3333	2003
Ironman	3	63.3333	2004
Ironman	4	71.6667	2003
Ironman	4	61.3333	2004
Ironman	5	54.6667	2003
Ironman	5	39.6667	2004
+----------+------------+---------+------+
50 rows in set (0.00 sec)

Write SQL queries for the below questions after loading the sample exercise data. All these queries should be joined with the "students" table to display the student name instead of student_id.

1. List the students who didn't appear for any exams.
Format: name

```
mysql> select s.name from students s, marks m where s.id=m.student_id and (quarterly is NULL and half_yearly is NULL and annual is NULL);
+------+
| name |
+------+
| Thor |
+------+
1 row in set (0.00 sec)
```

2. Find the total marks scored by each students in the annual exams. If the student hasn't appeared for any annual exam, he should still be listed with total marks scored as "0".
Format: name, marks, year

```
mysql> select name ,coalesce(sum(annual),0) marks,year from students s inner join marks m on s.id=m.student_id group by m.student_id,m.year;
+-----------+-------+------+
| name | marks | year |
+-----------+-------+------+
| Thor | 424 | 2003 |
| Thor | 309 | 2004 |
| Thor | 375 | 2005 |
| Thor | 296 | 2006 |
| Thor | 212 | 2007 |
| Hulk | 296 | 2003 |
| Hulk | 463 | 2004 |
| Hulk | 295 | 2005 |
```

Hulk	218	2006
Hulk	429	2007
Daredevil	465	2003
Daredevil	319	2004
Daredevil	385	2005
Daredevil	318	2006
Daredevil	306	2007
X Man	261	2003
X Man	307	2004
X Man	312	2005
X Man	287	2006
X Man	354	2007
Ironman	323	2003
Ironman	287	2004
Ironman	287	2005
Ironman	320	2006
Ironman	412	2007
+-----------+-------+------+
25 rows in set (0.00 sec)

3. List the students with the total marks scored in quarterly from all the subjects they had appeared during the year 2003.
Format: name, total, grade

mysql> select name, sum(coalesce(quarterly,0)) total,grade from students s inner join marks m on s.id=m.student_id and year='2003' group by m.student_id;
+-----------+-------+-------+
| name | total | grade |
+-----------+-------+-------+
Thor	0	6
Hulk	203	6
Daredevil	428	6
X Man	300	6
Ironman	277	6
+-----------+-------+-------+
5 rows in set (0.00 sec)

4. List the 9th and 10th grade students who received more than 3 medals.
Format: name, grade, no_of_medals

```
select name,m.grade,count(me.student_id) no_of_medals from students s left join
      (select student_id from medals) me on s.id=me.student_id left join (select
student_id,grade from marks where grade=10 or grade=9 group by  student_id, grade)
m on s.id=m.student_id group by  m.grade,s.id having  no_of_medals>3;
+-----------+-------+--------------+
| name | grade | no_of_medals |
+-----------+-------+--------------+
| Thor | 9 | 4 |
| Hulk | 9 | 4 |
| Daredevil | 9 | 5 |
| Thor | 10 | 4 |
| Hulk | 10 | 4 |
| Daredevil | 10 | 5 |
+-----------+-------+--------------+
6 rows in set (0.00 sec)
```

5. List the students who got less than 2 medals. This should also include the students who has not won any medals.
Format: name, grade, no_of_medals

```
mysql> select name,m.grade,count(me.student_id) no_of_medals from students s left
join (select student_id from medals) me on s.id=me.student_id left join (select
student_id,grade from marks where grade=10 or grade=9 group by student_id,grade) m
on s.id=m.student_id group by m.grade,s.id having no_of_medals<2;
+-------------+-------+--------------+
| name | grade | no_of_medals |
+-------------+-------+--------------+
| Quicksilver | NULL | 0 |
| X Man | 9 | 0 |
| Ironman | 9 | 0 |
| X Man | 10 | 0 |
| Ironman | 10 | 0 |
+-------------+-------+--------------+
5 rows in set (0.00 sec)
```

6. List the students who has not yet received any medals but scored more than 90 marks in all the subjects in the annual exam for that year.
Format: name, year

```
mysql> select name, m.year from students s,marks m where s.id=m.student_id and m.student_id not in(select student_id from medals ) group by m.student_id,year having min(annual)>90;
Empty set (0.00 sec)
```

7. List the records from the medals table for the students who had won more than 3 medals.
Format: name, game_id, medal_won, year, grade

```
mysql> select name,game_id,medal_won,year,grade from students s,medals m where s.id=m.student_id and s.id in(select student_id from medals group by student_id having count(*)>3);
+-----------+---------+-----------+------+-------+
| name | game_id | medal_won | year | grade |
+-----------+---------+-----------+------+-------+
| Daredevil | 5 | gold | 2003 | 6 |
| Thor | 5 | silver | 2003 | 6 |
| Hulk | 4 | silver | 2003 | 6 |
| Daredevil | 1 | gold | 2003 | 6 |
| Daredevil | 3 | bronze | 2004 | 7 |
| Daredevil | 4 | silver | 2004 | 7 |
| Hulk | 2 | silver | 2004 | 7 |
| Hulk | 4 | bronze | 2004 | 7 |
| Hulk | 5 | gold | 2004 | 7 |
| Daredevil | 3 | bronze | 2005 | 8 |
| Thor | 2 | gold | 2005 | 8 |
| Thor | 3 | bronze | 2005 | 8 |
| Thor | 4 | bronze | 2005 | 8 |
+-----------+---------+-----------+------+-------+
13 rows in set (0.00 sec)
```

8.List the number of medals and percentage of marks(based on total for the 5 subjects) scored in each year.
Format: name, medals, quarterly_per, half_yearly_per, annual_per, year, grade

```
mysql> select name, coalesce(medals,0),quarterly_per,half_yearly_per,annual_per,
a.year,b.grade from(select student_id,sum(quarterly)/5 quarterly_per,sum(half_yearly)/5
half_yearly_per,sum(annual)/5 annual_per,year from marks group by student_id,year) a
left join (select student_id,grade,count(*) medals, year from medals group by
student_id,year) b on a.student_id=b.student_id and a.year=b.year inner join students s
on s.id=a.student_id;
+-----------+-------------------+--------------+----------------+------------+------+-------+
| name | coalesce(medals,0) | quarterly_per | half_yearly_per | annual_per | year | grade
|
+-----------+-------------------+--------------+----------------+------------+------+-------+
| vishal | 1 | 0.0000 | 45.2000 | 84.8000 | 2003 | 6 |
| vishal | 0 | 47.2000 | 58.8000 | 61.8000 | 2004 | NULL |
| vishal | 3 | 66.8000 | 71.0000 | 75.0000 | 2005 | 8 |
| vishal | 0 | 57.6000 | 63.8000 | 59.2000 | 2006 | NULL |
| vishal | 0 | 60.2000 | 58.4000 | 42.4000 | 2007 | NULL |
| vishal | 0 | 16.0000 | 15.8000 | 18.2000 | 2013 | NULL |
| vishal | 0 | 18.0000 | 15.8000 | 18.2000 | 2014 | NULL |
| Hulk | 1 | 40.6000 | 51.4000 | 59.2000 | 2003 | 6 |
| Hulk | 3 | 64.8000 | 58.2000 | 92.6000 | 2004 | 7 |
| Hulk | 0 | 65.6000 | 54.8000 | 59.0000 | 2005 | NULL |
| Hulk | 0 | 73.8000 | 49.6000 | 43.6000 | 2006 | NULL |
| Hulk | 0 | 60.6000 | 54.0000 | 85.8000 | 2007 | NULL |
| Daredevil | 2 | 85.6000 | 75.0000 | 93.0000 | 2003 | 6 |
| Daredevil | 2 | 64.8000 | 62.8000 | 63.8000 | 2004 | 7 |
| Daredevil | 1 | 66.8000 | 49.8000 | 77.0000 | 2005 | 8 |
| Daredevil | 0 | 81.0000 | 86.8000 | 63.6000 | 2006 | NULL |
| Daredevil | 0 | 79.8000 | 73.8000 | 61.2000 | 2007 | NULL |
| X Man | 0 | 60.0000 | 62.0000 | 52.2000 | 2003 | NULL |
| X Man | 0 | 75.6000 | 69.0000 | 61.4000 | 2004 | NULL |
| X Man | 0 | 74.8000 | 65.2000 | 62.4000 | 2005 | NULL |
| X Man | 0 | 76.2000 | 71.0000 | 57.4000 | 2006 | NULL |
| X Man | 0 | 56.6000 | 48.2000 | 70.8000 | 2007 | NULL |
| Ironman | 0 | 55.4000 | 56.8000 | 64.6000 | 2003 | NULL |
| Ironman | 0 | 58.6000 | 55.4000 | 57.4000 | 2004 | NULL |
| Ironman | 0 | 51.0000 | 77.0000 | 57.4000 | 2005 | NULL |
```

```
| Ironman | 0 | 86.2000 | 68.8000 | 64.0000 | 2006 | NULL |
| Ironman | 0 | 74.2000 | 77.2000 | 82.4000 | 2007 | NULL |
+----------+------------------+--------------+----------------+-----------+------+-------+
27 rows in set (0.00 sec)
```

9. Lets assign some rating for the total marks scored - S(450-500), A(400-449), B(350-399), C(300-349), D(250,299), E(200-249), F(below 200). List the students with the grade obtained in each year for each exam(quarterly, half-yearly and annual)
Format: name, quarterly_rating, half_yearly_rating, annual_rating, year

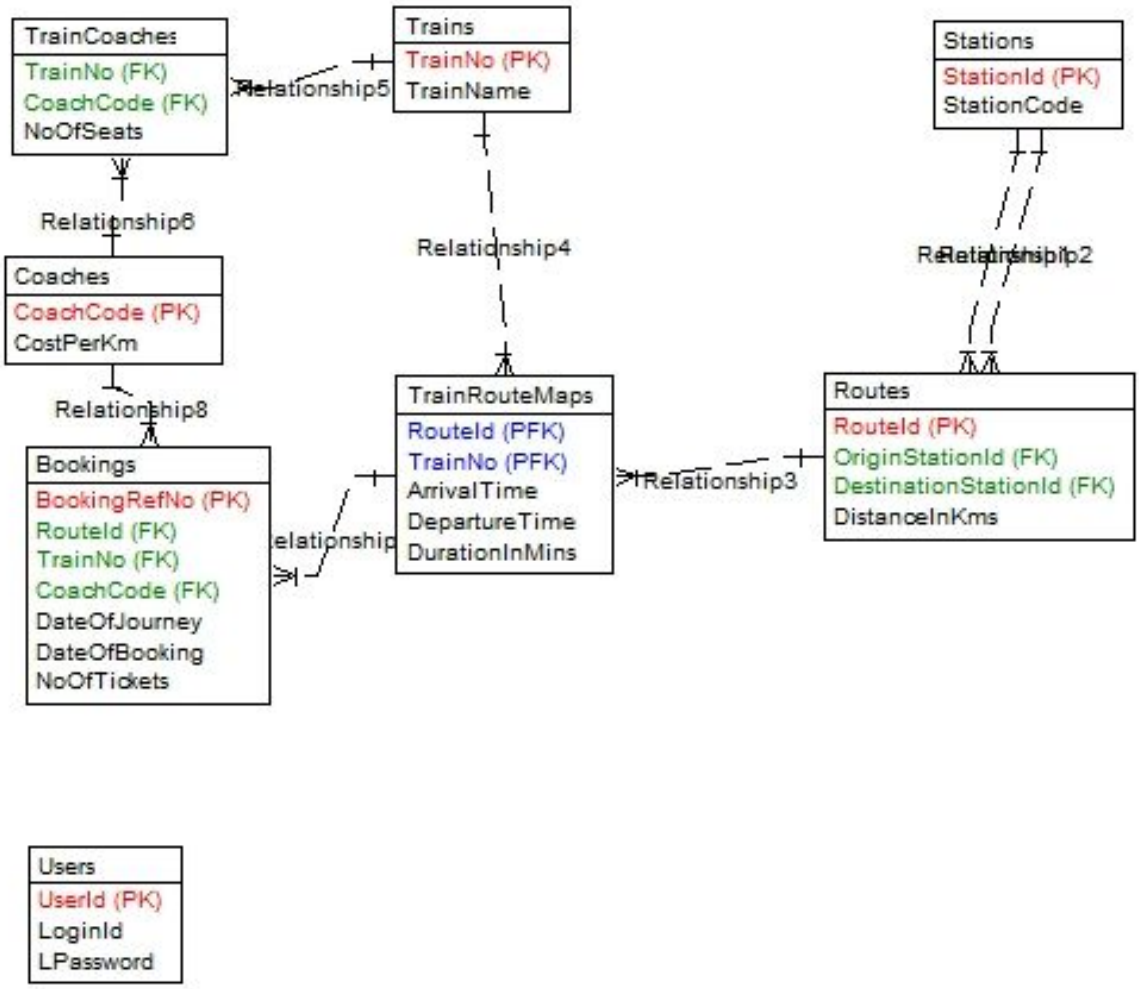
```
mysql> select name,
if(sum(quarterly)>=450,'S',
if(sum(quarterly)>=400,'A',
if(sum(quarterly)>=350,'B',
if(sum(quarterly)>=300,'C',
if(sum(quarterly)>=250,'D',
if(sum(quarterly)>=200,'E',
'F'))))))  quarterly_rating,
if(sum(half_yearly)>=450,'S',
if(sum(half_yearly)>=400,'A',
if(sum(half_yearly)>=350,'B',
if(sum(half_yearly)>=300,'C',
if(sum(half_yearly)>=250,'D',
if(sum(half_yearly)>=200,'E',
'F'))))))  half_yearly_rating ,
if(sum(annual)>=450,'S',
if(sum(annual)>=400,'A',
if(sum(annual)>=350,'B',
if(sum(annual)>=300,'C',
if(sum(annual)>=250,'D',
if(sum(annual)>=200,'E',
'F'))))))  annual_rating ,year from students s,marks m where s.id=m.student_id group by
student_id,year;
+-----------+-----------------+-------------------+---------------+------+
| name | quarterly_rating | half_yearly_rating | annual_rating | year |
+-----------+-----------------+-------------------+---------------+------+
```

Thor	F	E	A	2003
Thor	E	D	C	2004
Thor	C	B	B	2005
Thor	D	C	D	2006
Thor	C	D	E	2007
Hulk	E	D	D	2003
Hulk	C	D	S	2004
Hulk	C	D	D	2005
Hulk	B	E	E	2006
Hulk	C	D	A	2007
Daredevil	A	B	S	2003
Daredevil	C	C	C	2004
Daredevil	C	E	B	2005
Daredevil	A	A	C	2006
Daredevil	B	B	C	2007
X Man	C	C	D	2003
X Man	B	C	C	2004
X Man	B	C	C	2005
X Man	B	B	D	2006
X Man	D	E	B	2007
Ironman	D	D	C	2003
Ironman	D	D	D	2004
Ironman	D	B	D	2005
Ironman	A	C	C	2006
Ironman	B	B	A	2007
+-----------+-----------------+-------------------+--------------+------+
25 rows in set (0.00 sec)

Exercise

- **Create table and table elements for this model and list the queries used.**

[1,1]



Users Table

mysql> create table users(userid varchar(20) primary key, loginid varchar(20),lpassword varchar(20));
Query OK, 0 rows affected (0.10 sec)

insert into users values('10001','vishal','kumar');
insert into users values('10002','raju','kumar');

```
insert into users values('10003','ajit','kumar');
insert into users values('10004','vaibhav','kumar');
```

Stations Table

```
mysql> create table stations(station_id varchar(20) primary key, station_code
varchar(20));
Query OK, 0 rows affected (0.09 sec)

insert into stations values('s0001','mas');
insert into stations values('s0002','pnbe');
insert into stations values('s0003','ndls');
insert into stations values('s0004','hwh');
insert into stations values('s0005','bbs');
insert into stations values('s0006','gaya');
insert into stations values('s0007','bngr');
```

Trains Table

```
mysql> create table trains(train_no varchar(20) primary key, train_name varchar(40));
Query OK, 0 rows affected (0.09 sec)

insert into trains values('1123','hawrah mail express');
insert into trains values('1124','sangmitra express');
insert into trains values('1125','coromandal express');
insert into trains values('1126','banglor express');
insert into trains values('1127','gaya express');
```

Coaches Table

```
mysql> create table coaches(coach_code varchar(20) primary key, cost_per_km
numeric(5,2));
Query OK, 0 rows affected (0.11 sec)

insert into coaches values('sleeper',5);
insert into coaches values('first ac',10);
insert into coaches values('second ac',8);
insert into coaches values('third ac',7);
insert into coaches values('chair',4);
insert into coaches values('general',2);
```

TrainCoaches Table

```
mysql> create table traincoaches(train_no varchar(20),coach_code
varchar(20),no_of_seats int, FOREIGN KEY(train_no) REFERENCES trains(train_no),
FOREIGN KEY(coach_code) REFERENCES coaches(coach_code));
Query OK, 0 rows affected (0.16 sec)

insert into traincoaches values(1123,'sleeper',100);
insert into traincoaches values(1123,'third ac',50);
insert into traincoaches values(1123,'general',400);
insert into traincoaches values(1124,'sleeper',200);
insert into traincoaches values(1124,'third ac',50);
insert into traincoaches values(1124,'second ac',50);
insert into traincoaches values(1125,'sleeper',400);
insert into traincoaches values(1125,'first ac',50);
insert into traincoaches values(1125,'second ac',50);
insert into traincoaches values(1126,'first ac',100);
insert into traincoaches values(1126,'third ac',200);
insert into traincoaches values(1126,'second ac',100);
insert into traincoaches values(1127,'chair',500);
```

Routes Table

```
mysql> create table routes(route_id varchar(20) primary key,originalstation_id
varchar(20),destinationstation_id varchar(20),distanceinkm int, FOREIGN
KEY(originalstation_id) REFERENCES stations(station_id), FOREIGN
KEY(destinationstation_id) REFERENCES stations(station_id));
Query OK, 0 rows affected (0.12 sec)

insert into routes values('r0001','s0001','s0002',2200);
insert into routes values('r0002','s0001','s0003',2100);
insert into routes values('r0003','s0001','s0004',1600);
insert into routes values('r0004','s0001','s0005',1200);
insert into routes values('r0005','s0001','s0006',2000);
insert into routes values('r0006','s0001','s0007',500);
insert into routes values('r0007','s0007','s0001',500);
```

TrainRouteMaps Table

```
mysql> create table trainroutemaps(route_id varchar(20) ,train_no
varchar(20),arrivaltime varchar(10),departuretime varchar(10),durationinmins
numeric(5), FOREIGN KEY(route_id) REFERENCES routes(route_id), FOREIGN
KEY(train_no) REFERENCES trains(train_no),primary key(route_id,train_no));
Query OK, 0 rows affected (0.12 sec)

insert into trainroutemaps values('r0001','1124','16:20','08:35',2740);
insert into trainroutemaps values('r0003','1123','18:20','04:35',2040);
```

```
insert into trainroutemaps values('r0003','1125','09:20','12:35',1945);
insert into trainroutemaps values('r0004','1123','18:20','11:25',1540);
insert into trainroutemaps values('r0004','1125','09:20','08:35',1440);
insert into trainroutemaps values('r0005','1127','15:20','07:35',2840);
insert into trainroutemaps values('r0006','1126','01:40','06:35',400);
insert into trainroutemaps values('r0007','1126','03:40','00:35',400);
```

Booking Table

```
mysql> create table bookings(bookingref_no varchar(20) primary key,route_id
varchar(20) ,train_no varchar(20),coach_code varchar(20),dateofjourney
date,dateofbooking date,nooftickets int, FOREIGN KEY(route_id) REFERENCES
routes(route_id), FOREIGN KEY(train_no) REFERENCES trains(train_no),FOREIGN
KEY(coach_code) REFERENCES coaches(coach_code));
Query OK, 0 rows affected (0.19 sec)

insert into bookings values('b00001','r0001','1124','sleeper','2006-02-15','2006-02-10',1);
insert into bookings values('b00002','r0001','1124','second
ac','2006-02-15','2006-02-10',5);
insert into bookings values('b00003','r0001','1124','second
ac','2006-02-15','2006-02-10',3);
insert into bookings values('b00004','r0003','1123','sleeper','2008-02-15','2008-02-11',1);
insert into bookings values('b00006','r0006','1125','second
ac','2010-04-16','2010-03-10',3);
insert into bookings values('b00007','r0004','1123','sleeper','2009-11-12','2009-10-11',1);
```

1. To get the list of all trains

```
mysql> select train_name from trains;
+--------------------+
| train_name |
+--------------------+
| hawrah mail express |
| sangmitra express |
| coromandal express |
| banglor express |
| gaya express |
+--------------------+
5 rows in set (0.00 sec)
```

2. To get the list of all train routes in the database

```
mysql> select t.train_name, s.station_code source_station,s1.station_code
destinstion_station from routes r left join trainroutemaps tm on tm.route_id=r.route_id
left join trains t on tm.train_no=t.train_no left join stations s on
r.originalstation_id=s.station_id left join stations s1 on
s1.station_id=r.destinationstation_id group by originalstation_id,destinationstation_id;
+--------------------+----------------+--------------------+
| train_name | source_station | destinstion_station |
+--------------------+----------------+--------------------+
| sangmitra express | mas | pnbe |
| NULL | mas | ndls |
| hawrah mail express | mas | hwh |
| hawrah mail express | mas | bbs |
| gaya express | mas | gaya |
| banglor express | mas | bngr |
+--------------------+----------------+--------------------+
6 rows in set (0.00 sec)
```

3. Total number of seats in the each train.

```
mysql> select train_name ,sum(no_of_seats) total_seats from trains t,traincoaches tc
where t.train_no=tc.train_no group by tc.train_no;
+--------------------+-------------+
| train_name | total_seats |
+--------------------+-------------+
| hawrah mail express | 550 |
| sangmitra express | 300 |
| coromandal express | 500 |
| banglor express | 400 |
| gaya express | 500 |
+--------------------+-------------+
5 rows in set (0.00 sec)
```

4. List of all routes goes to Bangalore

```
mysql> select route_id, s.station_code,s1.station_code from routes r inner join stations
s on r.originalstation_id=s.station_id inner join stations s1 on
s1.station_id=r.destinationstation_id and s1.station_code='bngr' group by
originalstation_id,destinationstation_id;
```

```
+----------+--------------+--------------+
| route_id | station_code | station_code |
+----------+--------------+--------------+
| r0006 | mas | bngr |
+----------+--------------+--------------+
1 row in set (0.00 sec)
```

5.List of all routes starting from bangalore, calcutta and chennai

```
mysql> select route_id, s.station_code,s1.station_code from routes r inner join stations
s on r.originalstation_id=s.station_id inner join stations s1 on
s1.station_id=r.destinationstation_id and s.station_code in('mas','hwh','bngr') group by
originalstation_id,destinationstation_id;
+----------+--------------+--------------+
| route_id | station_code | station_code |
+----------+--------------+--------------+
| r0001 | mas | pnbe |
| r0002 | mas | ndls |
| r0003 | mas | hwh |
| r0004 | mas | bbs |
| r0005 | mas | gaya |
| r0006 | mas | bngr |
| r0007 | bngr | mas |
+----------+--------------+--------------+
7 rows in set (0.00 sec)
```

6.List of all the bookings booked between 1st Jan 2005 and 31st Dec 2005

```
mysql> select * from bookings where dateofbooking between '2001-01-01' and
'2012-01-01';
+--------------+----------+----------+------------+--------------+--------------+-------------+
| bookingref_no | route_id | train_no | coach_code | dateofjourney | dateofbooking |
nooftickets |
+--------------+----------+----------+------------+--------------+--------------+-------------+
| b00001 | r0001 | 1124 | sleeper | 2006-02-15 | 2006-02-10 | 1 |
| b00002 | r0001 | 1124 | second ac | 2006-02-15 | 2006-02-10 | 5 |
| b00003 | r0001 | 1124 | second ac | 2006-02-15 | 2006-02-10 | 3 |
| b00004 | r0003 | 1123 | sleeper | 2008-02-15 | 2008-02-11 | 1 |
| b00005 | r0003 | 1125 | second ac | 2006-04-16 | 2007-03-10 | 3 |
| b00006 | r0006 | 1125 | second ac | 2010-04-16 | 2010-03-10 | 3 |
```

| b00007 | r0004 | 1123 | sleeper | 2009-11-12 | 2009-10-11 | 1 |
+--------------+----------+----------+------------+--------------+--------------+-------------+
7 rows in set (0.00 sec)

7. List of all trains whose name begins with B

mysql> select train_name from trains where train_name like 'b%'
-> ;
+-----------------+
| train_name |
+-----------------+
| banglor express |
+-----------------+
1 row in set (0.00 sec)

8. List of all bookings where DOB is not null

mysql> select * from bookings where dateofbooking is not NULL;
+--------------+----------+----------+------------+--------------+--------------+-------------+
| bookingref_no | route_id | train_no | coach_code | dateofjourney | dateofbooking |
nooftickets |
+--------------+----------+----------+------------+--------------+--------------+-------------+
b00001	r0001	1124	sleeper	2006-02-15	2006-02-10	1
b00002	r0001	1124	second ac	2006-02-15	2006-02-10	5
b00003	r0001	1124	second ac	2006-02-15	2006-02-10	3
b00004	r0003	1123	sleeper	2008-02-15	2008-02-11	1
b00005	r0003	1125	second ac	2006-04-16	2007-03-10	3
b00006	r0006	1125	second ac	2010-04-16	2010-03-10	3
b00007	r0004	1123	sleeper	2009-11-12	2009-10-11	1
+--------------+----------+----------+------------+--------------+--------------+-------------+
7 rows in set (0.00 sec)

9. List of all bookings for the booked in 2006, DOJ in 2007

mysql> select * from bookings where year(dateofbooking)=2006 and
year(dateofjourney)=2007;
Empty set (0.00 sec)

10. Total number of coaches in the all the trains

```
mysql> select train_name,count(*) noofcoaches from trains t,traincoaches tc where
t.train_no=tc.train_no group by tc.train_no;
+--------------------+-------------+
| train_name | noofcoaches |
+--------------------+-------------+
| hawrah mail express | 3 |
| sangmitra express | 3 |
| coromandal express | 3 |
| banglor express | 3 |
| gaya express | 1 |
+--------------------+-------------+
5 rows in set (0.00 sec)
```

11.Total number of bookings in the database for the train no 1124

```
mysql> select train_no,count(train_no) no_of_booking from bookings where
train_no='1124' group by train_no;
+----------+---------------+
| train_no | no_of_booking |
+----------+---------------+
| 1124 | 3 |
+----------+---------------+
1 row in set (0.00 sec)
```

12. Total no of tickets column 'total' , booked for train no. 1124

```
mysql> select sum(nooftickets) total from bookings where train_no='1124';
+-------+
| total |
+-------+
| 9 |
+-------+
1 row in set (0.00 sec)
```

13. Minimum distance route

```
mysql> select s.station_code, s1.station_code from routes r left join stations s on
s.station_id=r.originalstation_id left join stations s1 on
```

s1.station_id=r.destinationstation_id group by originalstation_id,destinationstation_id having min(distanceinkm) ;

```
+-------------+-------------+
| station_code | station_code |
+-------------+-------------+
| mas | pnbe |
| mas | ndls |
| mas | hwh |
| mas | bbs |
| mas | gaya |
| mas | bngr |
+-------------+-------------+
6 rows in set (0.01 sec)
```

14. Total number of tickets booked for each train in the database

mysql> select train_name, sum(nooftickets) total_ticket_booked from trains t, bookings b where t.train_no=b.train_no group by b.train_no;

```
+--------------------+--------------------+
| train_name | total_ticket_booked |
+--------------------+--------------------+
| hawrah mail express | 2 |
| sangmitra express | 9 |
| coromandal express | 6 |
+--------------------+--------------------+
3 rows in set (0.00 sec)
```

15. cost for 50 kms for each coach.

mysql> select coach_code,cost_per_km*50 from coaches;

```
+------------+---------------+
| coach_code | cost_per_km*50 |
+------------+---------------+
| chair | 200.00 |
| first ac | 500.00 |
| general | 100.00 |
| second ac | 400.00 |
| sleeper | 250.00 |
| third ac | 350.00 |
+------------+---------------+
6 rows in set (0.01 sec)
```

16. List the train name and departure time for the trains starting from Bangalore.

```
mysql> select t.train_name,s.station_code, s1.station_code from routes r inner join
stations s on s.station_id=r.originalstation_id inner join trainroutemaps tm on
tm.route_id=r.route_id inner join trains t on tm.train_no=t.train_no inner join stations s1
on s1.station_id=r.destinationstation_id and s.station_code='bngr' group by
originalstation_id,destinationstation_id ;
+----------------+--------------+--------------+
| train_name | station_code | station_code |
+----------------+--------------+--------------+
| banglor express | bngr | mas |
+----------------+--------------+--------------+
1 row in set (0.00 sec)
```

17. List the trains for which the total no of tickets booked is greater than 500

```
mysql> select train_name from trains t,bookings b where t.train_no=b.train_no group by
b.train_no having sum(nooftickets)>500;
Empty set (0.00 sec)
```

18. List the trains for which the total no of tickets booked is lesser than 50

```
mysql> select train_name from trains t,bookings b where t.train_no=b.train_no group by
b.train_no having sum(nooftickets)<50;
+--------------------+
| train_name |
+--------------------+
| hawrah mail express |
| sangmitra express |
| coromandal express |
+--------------------+
3 rows in set (0.00 sec)
```

19. List the bookings along with train name, origin station, destination station and coach code after the date of journey ’25th Feb 2015’

```
mysql> select t.train_name, s.station_code,s1.station_code,coach_code from bookings
b inner join routes r on b.route_id=r.route_id inner join trains t on b.train_no=t.train_no
inner join stations s on r.originalstation_id=s.station_id inner join stations s1 on
s1.station_id=r.destinationstation_id and dateofbooking>'2001-01-01' group by
originalstation_id,destinationstation_id;
+--------------------+-------------+-------------+------------+
| train_name | station_code | station_code | coach_code |
+--------------------+-------------+-------------+------------+
| sangmitra express | mas | pnbe | sleeper |
| hawrah mail express | mas | hwh | sleeper |
| hawrah mail express | mas | bbs | sleeper |
| coromandal express | mas | bngr | second ac |
+--------------------+-------------+-------------+------------+
4 rows in set (0.00 sec)
```

20. List the trains via the route Mysore - Chennai

```
mysql> select t.train_name,s.station_code, s1.station_code from routes r inner join
stations s on s.station_id=r.originalstation_id inner join trainroutemaps tm on
tm.route_id=r.route_id inner join trains t on tm.train_no=t.train_no inner join stations s1
on s1.station_id=r.destinationstation_id and s.station_code='mas' and
s1.station_code='bngr' group by originalstation_id,destinationstation_id ;
+-----------------+-------------+-------------+
| train_name | station_code | station_code |
+-----------------+-------------+-------------+
| banglor express | mas | bngr |
+-----------------+-------------+-------------+
1 row in set (0.00 sec)
```

Day 4

1. Add columns 'created_at' and 'updated_at' to the tables marks, students and medals.

```
mysql> alter table students add column created_at timestamp default '0000-00-00
00:00:00',add column updated_at timestamp default now() on update now();
Query OK, 6 rows affected (0.28 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

```
mysql> alter table marks add column created_at timestamp default '0000-00-00
00:00:00',add column updated_at timestamp default now() on update now();
Query OK, 0 rows affected (0.26 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> alter table medals add column created_at timestamp default '0000-00-00
00:00:00',add column updated_at timestamp default now() on update now();
Query OK, 0 rows affected (0.26 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

2. Replace the null values in quarterly, half_yearly and annual columns with 0 and make those columns as not nullable.

```
mysql> UPDATE marks SET quarterly=0 WHERE quarterly IS NULL;
Query OK, 7 rows affected (0.05 sec)
Rows matched: 7 Changed: 7 Warnings: 0

mysql> UPDATE marks SET half_yearly=0 WHERE half_yearly IS NULL;
Query OK, 4 rows affected (0.06 sec)
Rows matched: 4 Changed: 4 Warnings: 0

mysql> UPDATE marks SET annual=0 WHERE annual IS NULL;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> alter table marks modify quarterly int(11) not null,modify half_yearly int(11) not
null,modify annual int(11) not null;
Query OK, 125 rows affected (0.26 sec)
Records: 125 Duplicates: 0 Warnings: 0
```

3. While inserting the value of updated_at & created_at should be the current time

```
INSERT INTO `students` (id,name,created_at)VALUES
(100007,'Thor',now()),(100008,'Hulk',now()),(100009,'Daredevil',now());
query OK, 3 rows affected (0.06 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> select * from students;+--------+-------------+---------------------+---------------------+
| id | name | created_at | updated_at |
+--------+-------------+---------------------+---------------------+
| 100001 | Thor | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| 100002 | Hulk | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| 100003 | Daredevil | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| 100004 | X Man | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| 100005 | Ironman | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| 100006 | Quicksilver | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| 100007 | Thor | 2015-12-09 19:00:54 | 2015-12-09 19:00:54 |
| 100008 | Hulk | 2015-12-09 19:00:54 | 2015-12-09 19:00:54 |
| 100009 | Daredevil | 2015-12-09 19:00:54 | 2015-12-09 19:00:54 |
+--------+-------------+---------------------+---------------------+
9 rows in set (0.00 sec)
```

4. While updating the value of updated_at alone should be the time of update

```
mysql> update students set name='vishal' where id='100001';
Query OK, 1 row affected (0.05 sec)
Rows matched: 1 Changed: 1 Warnings: 0


mysql> select * from students;
+--------+-------------+---------------------+---------------------+
| id | name | created_at | updated_at |
+--------+-------------+---------------------+---------------------+
| 100001 | vishal | 0000-00-00 00:00:00 | 2015-12-09 19:02:32 |
| 100002 | Hulk | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| 100003 | Daredevil | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| 100004 | X Man | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| 100005 | Ironman | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| 100006 | Quicksilver | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| 100007 | Thor | 2015-12-09 19:00:54 | 2015-12-09 19:00:54 |
| 100008 | Hulk | 2015-12-09 19:00:54 | 2015-12-09 19:00:54 |
```

| 100009 | Daredevil | 2015-12-09 19:00:54 | 2015-12-09 19:00:54 |
+--------+------------+---------------------+---------------------+
9 rows in set (0.00 sec)

Write a query for the problem given

Using the table training_sample, do the following

create a table students_summary with the below columns

- student_id
- student_name
- year
- percentage (got in annual exams)
- no_of_medals_received

Derive the values from the tables(students, marks and medals) and insert into the above table Use insert with select statement

mysql> select a.student_id,s.name,a.year,a.percentage,coalesce(no_of_medals,0) from(select student_id, year, sum(annual)/5 percentage from marks group by student_id,year) a left join (select student_id,year,count(*) no_of_medals from medals group by student_id,year) b on a.student_id=b.student_id and a.year=b.year inner join students s on a.student_id=s.id;+------------+-----------+------+------------+-------------------------+
| student_id | name | year | percentage | coalesce(no_of_medals,0) |
+------------+-----------+------+------------+-------------------------+
100001	vishal	2003	84.8000	1
100001	vishal	2004	61.8000	0
100001	vishal	2005	75.0000	3
100001	vishal	2006	59.2000	0
100001	vishal	2007	42.4000	0

100002	Hulk	2003	59.2000	1
100002	Hulk	2004	92.6000	3
100002	Hulk	2005	59.0000	0
100002	Hulk	2006	43.6000	0
100002	Hulk	2007	85.8000	0
100003	Daredevil	2003	93.0000	2
100003	Daredevil	2004	63.8000	2
100003	Daredevil	2005	77.0000	1
100003	Daredevil	2006	63.6000	0
100003	Daredevil	2007	61.2000	0
100004	X Man	2003	52.2000	0
100004	X Man	2004	61.4000	0
100004	X Man	2005	62.4000	0
100004	X Man	2006	57.4000	0
100004	X Man	2007	70.8000	0
100005	Ironman	2003	64.6000	0
100005	Ironman	2004	57.4000	0
100005	Ironman	2005	57.4000	0
100005	Ironman	2006	64.0000	0
100005	Ironman	2007	82.4000	0
+------------+-----------+------+------------+-------------------------+
25 rows in set (0.00 sec)

mysql> create table student_summary(student_id varchar(20),student_name varchar(100),year int(11),percentage numeric(5,4),no_of_medals_received int(11));
Query OK, 0 rows affected (0.12 sec)

mysql> insert into student_summary(student_id,student_name,year,percentage,no_of_medals_received) (select a.student_id,s.name,a.year,a.percentage,coalesce(no_of_medals,0) from(select student_id, year, sum(annual)/5 percentage from marks group by student_id,year) a left join (select student_id,year,count(*) no_of_medals from medals group by student_id,year) b on a.student_id=b.student_id and a.year=b.year inner join students s on a.student_id=s.id);
Query OK, 25 rows affected, 25 warnings (0.08 sec)
Records: 25 Duplicates: 0 Warnings:

Triggers

Exercises:

1. **Add a column average to the table marks. Write a trigger to calculate the average marks for the subject whenever any insert/update happens in the columns quarterly, half_yearly and annual in the table 'marks'.**

alter table add column average numeric(4,2);

```
mysql> create trigger update_avg before insert on marks
-> for each row
-> begin
-> set new.average=(new.quarterly+new.half_yearly+new.annual)/3;
-> end;
-> //
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> INSERT INTO
`marks`(student_id,subject_id,quarterly,half_yearly,annual,year,grade) VALUES
(100001,1,90,79,91,2014,6);//
Query OK, 1 row affected (0.05 sec)

mysql> select * from marks;
-> //
+------+------------+------------+-----------+-------------+--------+------+-------+---------------------+---
------------------+---------+
| id | student_id | subject_id | quarterly | half_yearly | annual | year | grade | created_at |
updated_at | average |
+------+------------+------------+-----------+-------------+--------+------+-------+---------------------+---
------------------+---------+
| 1001 | 100001 | 1 | 0 | 79 | 91 | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
NULL |
| 1002 | 100002 | 1 | 35 | 49 | 77 | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00
| NULL |
| 1003 | 100003 | 1 | 100 | 97 | 95 | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00
00:00:00 | NULL |
| 1004 | 100004 | 1 | 38 | 38 | 65 | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00
| NULL |
```

1005	100005	1	73	40	100	2003	6	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1006	100001	2	0	30	76	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1007	100002	2	37	45	87	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1008	100003	2	93	91	98	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1009	100004	2	93	59	63	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1010	100005	2	34	89	45	2003	6	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1011	100001	3	0	86	89	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1012	100002	3	46	76	48	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1013	100003	3	46	0	83	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1014	100004	3	71	74	31	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1015	100005	3	34	54	36	2003	6	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1016	100001	4	0	0	80	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1017	100002	4	52	43	31	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1018	100003	4	91	95	99	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1019	100004	4	46	76	39	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1020	100005	4	80	41	94	2003	6	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1021	100001	5	0	31	88	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1022	100002	5	33	44	53	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1023	100003	5	98	92	90	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1024	100004	5	52	63	63	2003	6	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1025	100005	5	56	60	48	2003	6	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1026	100001	1	59	34	57	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL

1027	100002	1	47	37	94	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1028	100003	1	47	80	97	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1029	100004	1	89	43	68	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1030	100005	1	72	82	47	2004	7	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1031	100001	2	44	54	31	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1032	100002	2	85	96	89	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1033	100003	2	84	63	57	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1034	100004	2	83	97	53	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1035	100005	2	53	30	80	2004	7	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1036	100001	3	44	90	54	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1037	100002	3	53	36	87	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1038	100003	3	64	55	39	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1039	100004	3	95	36	54	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1040	100005	3	66	87	37	2004	7	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1041	100001	4	31	65	69	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1042	100002	4	98	30	94	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1043	100003	4	43	60	83	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1044	100004	4	76	79	40	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1045	100005	4	66	43	75	2004	7	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1046	100001	5	58	51	98	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1047	100002	5	41	92	99	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1048	100003	5	86	56	43	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL

1049	100004	5	35	90	92	2004	7	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1050	100005	5	36	35	48	2004	7	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1051	100001	1	96	57	94	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1052	100002	1	68	41	50	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1053	100003	1	60	83	73	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1054	100004	1	53	81	97	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1055	100005	1	64	81	39	2005	8	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1056	100001	2	77	82	36	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1057	100002	2	73	57	52	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1058	100003	2	38	35	81	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1059	100004	2	69	46	44	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1060	100005	2	35	95	39	2005	8	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1061	100001	3	32	70	58	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1062	100002	3	43	48	65	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1063	100003	3	81	38	90	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1064	100004	3	88	90	92	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1065	100005	3	34	95	76	2005	8	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1066	100001	4	64	69	87	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1067	100002	4	62	38	95	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1068	100003	4	79	49	86	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1069	100004	4	68	33	33	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1070	100005	4	72	39	84	2005	8	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22

1071	100001	5	65	77	100	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1072	100002	5	82	90	33	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1073	100003	5	76	44	55	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1074	100004	5	96	76	46	2005	8	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1075	100005	5	50	75	49	2005	8	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1076	100001	1	67	84	51	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1077	100002	1	63	68	46	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1078	100003	1	89	96	31	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1079	100004	1	86	74	82	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1080	100005	1	91	49	70	2006	9	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1081	100001	2	65	90	67	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1082	100002	2	77	38	75	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1083	100003	2	100	99	95	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1084	100004	2	100	37	40	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1085	100005	2	85	69	37	2006	9	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1086	100001	3	0	0	32	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1087	100002	3	98	34	31	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1088	100003	3	65	83	36	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1089	100004	3	82	75	67	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1090	100005	3	93	78	39	2006	9	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1091	100001	4	83	75	59	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1092	100002	4	81	56	30	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL

1093	100003	4	94	60	68	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1094	100004	4	59	92	47	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1095	100005	4	76	82	83	2006	9	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1096	100001	5	73	70	87	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1097	100002	5	50	52	36	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1098	100003	5	57	96	88	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1099	100004	5	54	77	51	2006	9	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1100	100005	5	86	66	91	2006	9	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1101	100001	1	0	0	0	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1102	100002	1	93	31	79	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1103	100003	1	81	92	69	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1104	100004	1	32	31	76	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1105	100005	1	82	65	87	2007	10	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1106	100001	2	62	80	36	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1107	100002	2	34	49	87	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1108	100003	2	95	89	89	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1109	100004	2	50	46	76	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1110	100005	2	86	87	84	2007	10	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1111	100001	3	91	64	80	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1112	100002	3	52	60	94	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1113	100003	3	50	81	58	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1114	100004	3	65	39	46	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL

1115	100005	3	44	79	91	2007	10	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1116	100001	4	55	96	55	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1117	100002	4	82	67	94	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1118	100003	4	85	76	47	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1119	100004	4	79	89	89	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1120	100005	4	72	56	57	2007	10	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1121	100001	5	93	52	41	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1122	100002	5	42	63	75	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1123	100003	5	88	31	43	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1124	100004	5	57	36	67	2007	10	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
1125	100005	5	87	99	93	2007	10	0000-00-00 00:00:00	2015-12-10 12:25:32	12.22
1127	100001	1	80	79	91	2013	6	0000-00-00 00:00:00	2015-12-10 13:13:51	99.99
1128	100001	1	90	79	91	2014	6	0000-00-00 00:00:00	2015-12-10 13:20:34	86.67

127 rows in set (0.00 sec)

Rename the column name from medal_won to medal_received in the table medals.

1. Add a column medal_received to the table medals.

```
mysql> alter table medals add column medal_received varchar(10) default null;
-> //
Query OK, 13 rows affected (0.33 sec)
Records: 13 Duplicates: 0 Warnings:
```

2. Write a trigger to copy the values to both the columns(medal_won and medal_received) whenever any of these columns inserted/updated.

```
mysql> create trigger copy_medal BEFORE INSERT ON medals
-> FOR EACH ROW
```

```
-> BEGIN
-> IF new.medal_won is NULL THEN
-> SET new.medal_won = new.medal_received;
-> ELSE
-> SET new.medal_received = new.medal_won;
-> END IF;
-> END;
-> //
Query OK, 0 rows affected (0.06 sec)



mysql> INSERT INTO `medals`(student_id,game_id,medal_won,year,grade) VALUES
(100003,5,'gold',2013,6);
-> //
Query OK, 1 row affected (0.06 sec)

mysql> select * from medals;
-> //
+-----+------------+---------+-----------+------+-------+---------------------+---------------------+---------
-------+
| id | student_id | game_id | medal_won | year | grade | created_at | updated_at |
medal_received |
+-----+------------+---------+-----------+------+-------+---------------------+---------------------+---------
-------+
| 101 | 100003 | 5 | gold | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL |
| 102 | 100001 | 5 | silver | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL
|
| 103 | 100002 | 4 | silver | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL
|
| 104 | 100003 | 1 | gold | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL |
| 105 | 100003 | 3 | bronze | 2004 | 7 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
NULL |
| 106 | 100003 | 4 | silver | 2004 | 7 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL
|
| 107 | 100002 | 2 | silver | 2004 | 7 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL
|
| 108 | 100002 | 4 | bronze | 2004 | 7 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
NULL |
| 109 | 100002 | 5 | gold | 2004 | 7 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL |
| 110 | 100003 | 3 | bronze | 2005 | 8 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
NULL |
| 111 | 100001 | 2 | gold | 2005 | 8 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL |
| 112 | 100001 | 3 | bronze | 2005 | 8 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
NULL |
```

```
| 113 | 100001 | 4 | bronze | 2005 | 8 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
NULL |
| 115 | 100003 | 5 | gold | 2013 | 6 | 0000-00-00 00:00:00 | 2015-12-10 13:43:37 | gold |
+-----+------------+---------+-----------+------+-------+---------------------+---------------------+---------
-------+
14 rows in set (0.00 sec)

mysql> INSERT INTO `medals`(student_id,game_id,year,grade,medal_received)
VALUES (100003,5,2014,6,'silver');
-> //
Query OK, 1 row affected (0.06 sec)

mysql> select * from medals;
-> //
+-----+------------+---------+-----------+------+-------+---------------------+---------------------+---------
-------+
| id | student_id | game_id | medal_won | year | grade | created_at | updated_at |
medal_received |
+-----+------------+---------+-----------+------+-------+---------------------+---------------------+---------
-------+
| 101 | 100003 | 5 | gold | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL |
| 102 | 100001 | 5 | silver | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL
|
| 103 | 100002 | 4 | silver | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL
|
| 104 | 100003 | 1 | gold | 2003 | 6 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL |
| 105 | 100003 | 3 | bronze | 2004 | 7 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
NULL |
| 106 | 100003 | 4 | silver | 2004 | 7 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL
|
| 107 | 100002 | 2 | silver | 2004 | 7 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL
|
| 108 | 100002 | 4 | bronze | 2004 | 7 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
NULL |
| 109 | 100002 | 5 | gold | 2004 | 7 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL |
| 110 | 100003 | 3 | bronze | 2005 | 8 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
NULL |
| 111 | 100001 | 2 | gold | 2005 | 8 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | NULL |
| 112 | 100001 | 3 | bronze | 2005 | 8 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
NULL |
| 113 | 100001 | 4 | bronze | 2005 | 8 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
NULL |
| 115 | 100003 | 5 | gold | 2013 | 6 | 0000-00-00 00:00:00 | 2015-12-10 13:43:37 | gold |
| 116 | 100003 | 5 | silver | 2014 | 6 | 0000-00-00 00:00:00 | 2015-12-10 13:45:30 | silver
|
```

```
+-----+------------+---------+-----------+------+-------+--------------------+--------------------+---------
-------+
15 rows in set (0.00 sec)
```

3. Drop the column medal_won.

```
mysql> alter table medals drop column medal_won;
-> //
Query OK, 15 rows affected (0.36 sec)
Records: 15 Duplicates: 0 Warnings: 0
```