

Curtin University – Department of Computing

Assignment Cover Sheet / Declaration of Originality

Last name:	troyee	Student ID:	2663281
Other name(s):	Dhrubo jouti das		
Unit name:	Database Systems	Unit ID:	SYS2014/ISYS 5008
Lecturer / unit coordinator:	Dr. Nimalika Fernando	Tutor:	
Date of submission:	24/10/2024	Which Assignment?	Final

Declaration

I declare that:

- The above information is complete and accurate.
- The work I am submitting is entirely my own, except where clearly indicated otherwise and correctly referenced.
- I have taken (and will continue to take) all reasonable steps to ensure my work is not accessible to any other students who may gain unfair advantage from it.
- I have not previously submitted this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I understand that:

- Plagiarism and collusion are dishonest, and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I plagiarise or collude, I risk failing the unit with a grade of ANN (“Result Annulled due to Academic Misconduct”), which will remain permanently on my academic record. I also risk termination from my course and other penalties.
- Even with correct referencing, my submission will only be marked according to what I have done myself, specifically for this assessment. I cannot re-use the work of others, or my own previously submitted work, in order to fulfil the assessment requirements.
- It is my responsibility to ensure that my submission is complete, correct and not corrupted.

Signature : Dhrubo Troyee

Date of Signature: 24/10/2024

1. Introduction

The design and execution of a MySQL relational database system created for the 2024 Olympic Games are described in this report. This database provides a foundation for querying data in a variety of formats and supports advanced database operations including stored procedures and views. Its goal is to handle information on athletes, events, medallist, teams, total medals, and schedules, coach.

In the user guide it is mentioned that there are **5 SQL files (CreateTable.sql , Load-Data.sql , Query.sql, advanceQuery.sql , DeleteTable.sql)** to efficiently use the Olympic game database . Again, there are 7 csv files that contains information to loading data into respective table like (athlete.csv , coach.csv , team.csv , schedule.csv , medallist.csv, medal_total.csv and events.csv).Additionally, it also contains **4 python file** to be connected the MYSQL Server in a python environments and do some operation such as (**pythonConnect.py , eventdata.py , scheduleData.py insert.py**). I also create a command. Txt file to see all the command together.

With an emphasis on efficiency and flexibility, the project included data modelling, database implementation, data loading, and query creation. Moreover, MySQL 8.0.39 has been used to implement the database on a Linux host running Ubuntu that can be accessed via the VMware Horizontal View Client. The design, implementation, data loading procedures, query usage, and complex database capabilities like stored procedures and views will all be covered in this study. The study will also examine the difficulties encountered and make recommendations for modifications for upcoming system versions.

2. Design of Database

2.1 Entity Sets

ENTITY SET	KEY	OTHER ATTRIBUTES
COACH	<u>coach_code</u>	coach_name, ,gender , coach_function , country_code , country_long , disciplines, events
ATHLETE	<u>athlete_code</u>	name , name_short, name_tv, gender, country_code, country_long, disciplines, events
TEAM	<u>code</u>	Team_gender,country_code,country,country_long,disciplines, disciplines_code, events, num_athletes
TOTAL_MEDAL	<u>medal_id</u>	country_code, country_long, Gold_Medal, Silver_Medal, Bronze_medal, Total
EVENT	<u>event_id</u>	event, sport, sport_code
MEDALLIST	<u>medallist_id</u> , <u>REF KEY</u> <u>code_athletes</u> , <u>code_team</u> ,	medal_date, medal_type, medal_code, name, gender, country_code, country_long, team_gender, discipline, events, event_type, birth_date, is_medallist,
SCHEDULE	<u>schedule_id</u>	start_date, end_date, status, discipline, event, event_medal, phrase, gender, venue, venue_code

Relationship Sets

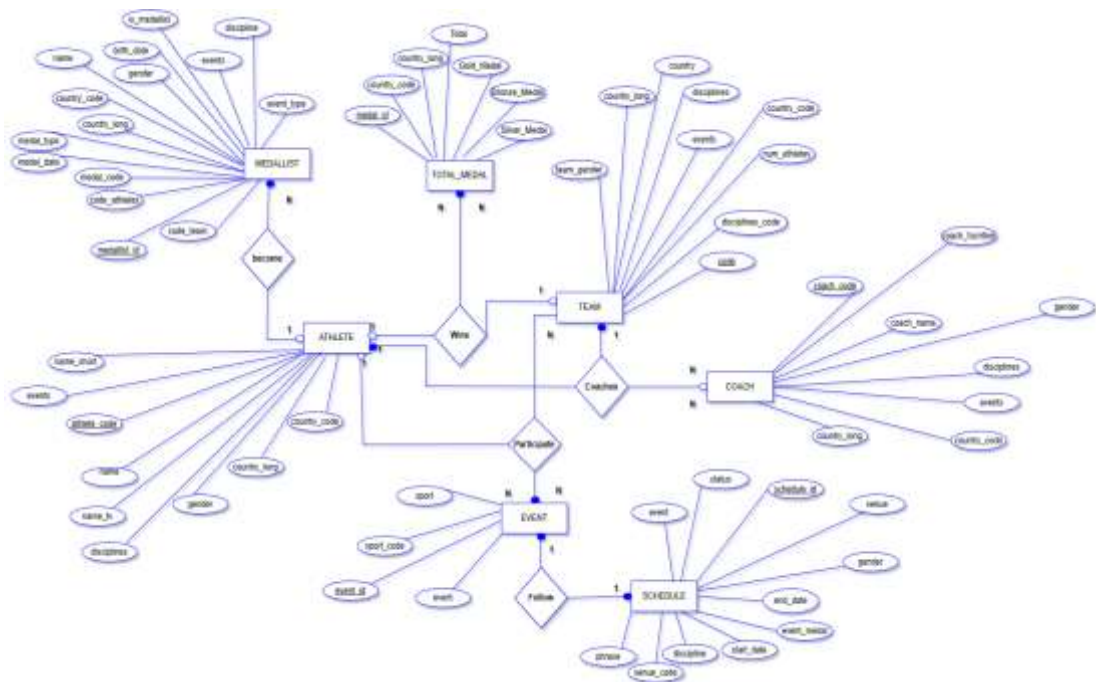
RELATIONSHIP SET	BETWEEN ENTITY SETS	ATTRIBUTES
BECOME WINS	ATHLETE, MEDALLIST ATHELTE, TOTAL_MEDAL	
WINS	TEAM, TOTAL_MEDAL	
COACHES COACHES	TEAM, COACH ATHLETE, COACH	
PARTICIPATE	TEAM, EVENT	
PARTICIPATE	ATHELET, EVENT	
FOLLOW	EVENT, SCHEDULE	

Constraints

Between Entity Sets	Relation-ship set	Cardinality	Participation
ATHLETE, MEDALLIST	become	One – Many Each athlete can win multiple medals (gold, siler, bronze) in various events. But each record refers to one athlete who won a medal.	ATHLETE - Partial MEDALLIST – Total An athlete can exit without being the medallist. But every medallist must be Athletes.
ATHELTE, TOTAL_MEDAL	wins	One – Many Each athlete can win multiple medals in various events. But each total medal record is associated with one athlete.	ATHLETE – Partial TOTAL_MEDAL - Total Not every athlete will win medals, so an athletes may exist without a medal. Every medal must be linked to an athlete who has won medals
TEAM, TOTAL_MEDAL	wins	One – Many A team can win multiple total medals in various events. But each total medal record is associated with one specific team.	TEAM – Partial, TOTAL_MEDAL - Total A team can exit without Winning any medals. Every medal must be belong to a team that won those medals.
TEAM, COACH	coaches	One -Many A team can have one coach. But a coach can coaches many team.	COACH – Partial, TEAM - Total Every team must have One coach. A coach can exit without being assigned to any team. But a team can not be exit without being trained from Coach.
COACH, ATHLETE,	coaches	One -Many	COACH – Partial, ATHLETE - Total

		A coach can be trains/ coaches multiple athletes. Each athlete is associated with one coach at a time.	A coach can exit without training to athletes. Every athletes must be assign to one coach for training.
TEAM, EVENT	participate	Many-Many One team can participant in multiple events But one eventscan have multiple teams	TEAM – Partial, EVENT - Partial A team can exit participating in an event An event (individual/team) can exit involving any team
ATHELET, EVENT	participate	One -Many An athlete can participate many events, But each event participa- tion linked to Many ath- lete	ATHELTE – Partial EVENT- Partial Not all athletes participate in every event. An event can occur without the participation of athletes.
EVENT, SCHEDULE	follow	One- One Each event must have one schedule. Each schedule is connected to only one event	EVENT – Total SCHEDULE – Total An event can not exit without having a schedule to take place. Also, an schedule can not be exit without an events.

2.2 ER Diagram



2.3 Data Description

ATHELTE_TABLE

Field	Type	Null	Key	Default	Extra
athlete_code	varchar(100)	NO	PRI	NULL	
name	char(70)	NO		NULL	
name_short	varchar(100)	NO		NULL	
name_tv	varchar(90)	NO		NULL	
gender	char(1)	NO		NULL	
country_code	char(3)	NO		NULL	
country_long	varchar(100)	NO		NULL	
disciplines	varchar(200)	NO		NULL	
events	varchar(100)	NO		NULL	

COACH_TABLE

Field	Type	Null	Key	Default	Extra
coach_code	varchar(100)	NO	PRI	NULL	
coach_name	char(70)	NO		NULL	
gender	char(3)	NO		NULL	
coach_function	varchar(50)	NO		NULL	
country_code	char(3)	NO		NULL	
country_long	varchar(100)	YES		NULL	
disciplines	varchar(200)	NO		NULL	
events	varchar(100)	YES		NULL	

TEAM_TABLE

Field	Type	Null	Key	Default	Extra
code	varchar(100)	NO	PRI	NULL	
team_gender	char(1)	NO		NULL	
country_code	char(3)	NO		NULL	
country	varchar(50)	NO		NULL	
country_long	varchar(100)	NO		NULL	
discipline	varchar(200)	NO		NULL	
disciplines_code	varchar(6)	NO		NULL	
events	varchar(100)	YES		NULL	
num_athletes	int	YES		NULL	

MEDALLIST_TABLE

Field	Type	Null	Key	Default	Extra
medal_date	date	YES		NULL	
medal_type	varchar(20)	YES		NULL	
medal_code	int	YES		NULL	
name	varchar(70)	YES		NULL	
gender	char(1)	YES		NULL	
country_code	char(3)	NO		NULL	
country_long	varchar(100)	NO		NULL	
team_gender	char(1)	YES		NULL	
discipline	varchar(200)	YES		NULL	
events	varchar(100)	YES		NULL	
event_type	varchar(20)	YES		NULL	
birth_date	date	YES		NULL	
code_athlete	varchar(100)	YES	MUL	NULL	
code_team	varchar(100)	YES	MUL	NULL	
is_medallist	tinyint(1)	YES		NULL	
medalist_id	int	NO	PRI	NULL	auto_increment

EVENT_TABLE

Field	Type	Null	Key	Default	Extra
event	varchar(100)	NO		NULL	
sport	varchar(200)	NO		NULL	
sport_code	varchar(10)	NO		NULL	
event_id	int	NO	PRI	NULL	auto_increment

SCHEDULE_TABLE

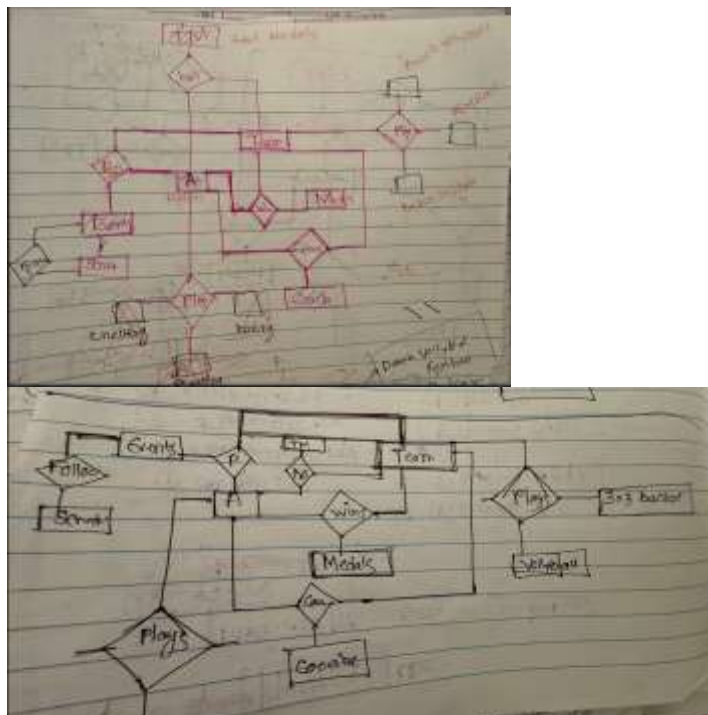
Field	Type	Null	Key	Default	Extra
start_date	timestamp	YES		NULL	
end_date	timestamp	YES		NULL	
status	varchar(20)	NO		NULL	
discipline	varchar(200)	NO		NULL	
discipline_code	varchar(60)	NO		NULL	
event	varchar(100)	YES		NULL	
event_medal	int	YES		NULL	
phase	varchar(100)	YES		NULL	
gender	char(1)	YES		NULL	
event_type	varchar(20)	YES		NULL	
venue	varchar(100)	YES		NULL	
venue_code	varchar(60)	YES		NULL	
schedule_id	int	NO	PRI	NULL	auto_increment

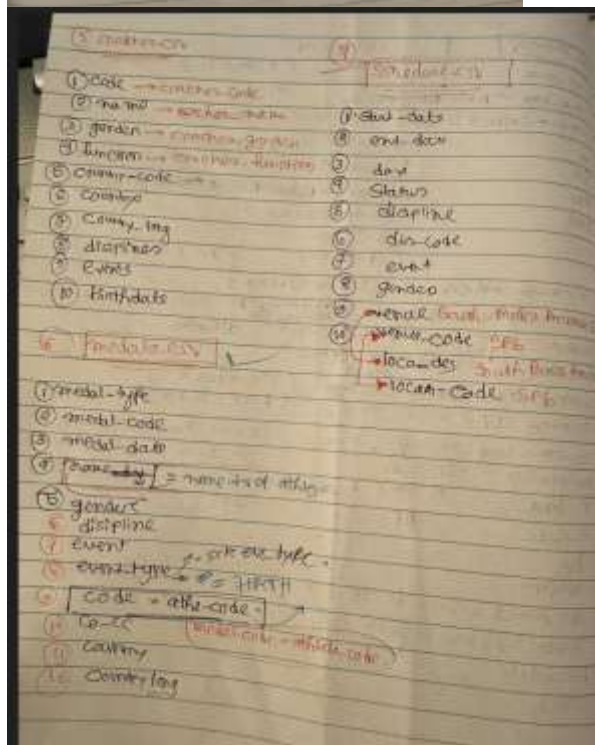
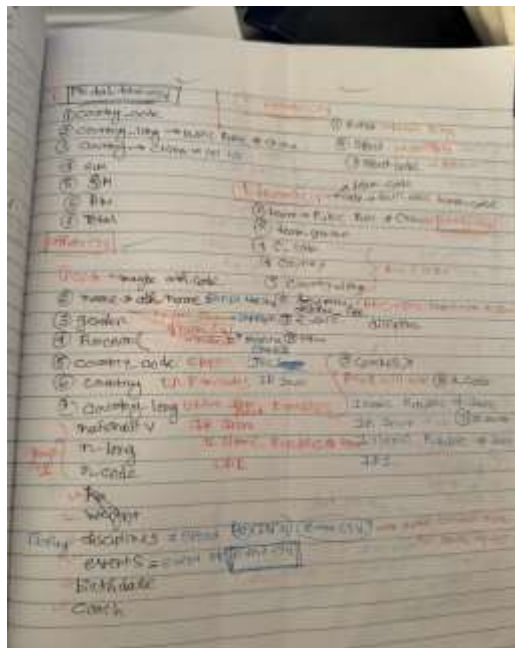
TOTAL_MEDAL TABLE

Field	Type	Null	Key	Default	Extra
country_code	char(3)	NO		NULL	
country_long	varchar(100)	NO		NULL	
Gold_Medal	int	YES		NULL	
Silver_Medal	int	YES		NULL	
Bronze_Medal	int	YES		NULL	
Total	int	YES		NULL	
medal_id	int	NO	PRI	NULL	auto_increment

2.4 Assumptions

I did a draft from the csv file that which attribute will I use to creating the tables and try to draw the ER diagram first, after finishing the draft. I changed it so many times, to get my desire ER diagram, and attributes. Here is a sample from one of it.





3. Implementation of the Database and Adding Sample Data

3.1 Table Creation and Constraints

The tables were created in the **CretaeTable.sql** scripts. There are constraints such as primary key foreign key, NOT NULL, NULL based on the data. Here is the file looks like



```
1 /* CreateTable.sql is the file for creating tables for Olympic Games 2024 database details.
2 HERE, 7 types of Tables are created .
3
4 1.COACH TABLE -- This table contains the details of coaches .
5 2. ATHLETE TABLE -- details of athletes of the game
6 3. TEAM TABLE -- details of team game details
7 4. TOTAL_MEDAL -- details of total medal of the game
8 5. EVENT TABLE -- contains details of event of the game
9 6. MEDALLIST TABLE -- contains the details of all medallist
10 7. SCHEDULE TABLE -- contains the details of event schedule of all game
11 */
12
13
14
15 -- 1. Creating COACH TABLE
16
17 DROP TABLE IF EXISTS COACH;
18
19 CREATE TABLE COACH (
20     coach_code VARCHAR(100), -- code of coach in COACH TABLE
21     coach_name CHAR(70) NOT NULL, -- name of coach
22     gender CHAR(1) NOT NULL, -- coach gender
23     coach_function VARCHAR(50) NOT NULL, -- coach function (eg : head,
        assistant )
24     country_code CHAR(3) NOT NULL, -- country of the code
25     country_long VARCHAR(100), -- full country name
26     disciplines VARCHAR(200) NOT NULL, -- name of games/ discipline
27 )
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
261
```


3.3 File Descriptions

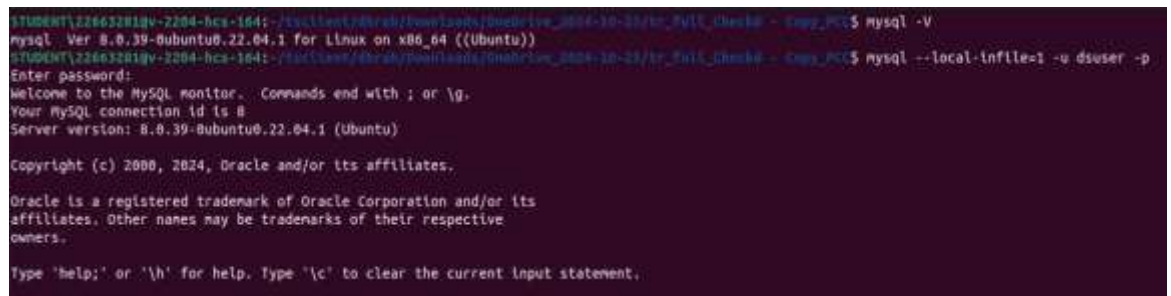
Each file has the explanation about the file description. Like in **Query.sql** have the description of about the file. Like this SQL file is about creating queries using *join*, *subqueries*, *aggregation function* and so on.



```
1 /* Query.sql ahve all the simple queies, join , using aggregation fucntion, sub queries */
2 -- Capturing all the output in a file
3 tee SimpleQueryAnswerOlympic.out
4
5
6 -- show available databases---
7 show databases;
8
9 -- use dswork databases
10
11
12 -- display the structure of all tables using DESC
13 DESC ATHLETE ;
14 DESC COACH;
15 DESC TEAM;
16 DESC MEDALLIST;
17 DESC EVENT;
18 DESC SCHEDULE;
19 DESC TOTAL_MEDAL;
20
21
22 -- Display all retrieve data using SELECT * from EACH table
23 SELECT * FROM ATHLETE;
24 SELECT * FROM TEAM;
25 SELECT * FROM COACH;
26 SELECT * FROM MEDALLIST;
27 SELECT * FROM EVENT;
28 SELECT * FROM SCHEDULE;
29 SELECT * FROM TOTAL_MEDAL;
30
```

4. Use of the Database

1. First, we need to login in VMware horizon client like this



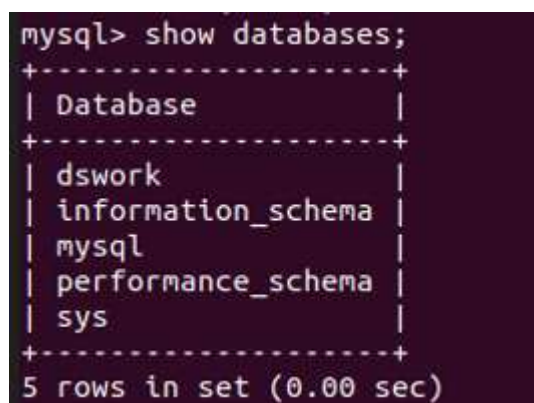
```
STUDENT\22663281qv-2204-hcs-164:~/tsclient/Downloads/OneDrive_2024-10-23/ry_full_Checkid - Copy_POC$ mysql -V
mysql Ver 8.0.39-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
STUDENT\22663281qv-2204-hcs-164:~/tsclient/Downloads/OneDrive_2024-10-23/ry_full_Checkid - Copy_POC$ mysql --local-infile=1 -u dsuser -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.39-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

2. Then see the existence DATABASE like this



```
mysql> show databases;
+-----+
| Database |
+-----+
| dswork   |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.00 sec)
```

3. Now need to create out own database which is;

```
mysql> CREATE DATABASE IF NOT EXISTS Olympic_Game_2024_22663281;
Query OK, 1 row affected (0.05 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| Olympic_Game_2024_22663281 |
| dswork |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0.00 sec)

mysql> use Olympic_Game_2024_22663281;
Database changed
```

- Now create all the tables using **Createtable.sql**

```
mysql> \. Createtable.sql
Query OK, 0 rows affected, 1 warning (0.01 sec)

Query OK, 0 rows affected (0.13 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.07 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected, 4 warnings (0.09 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.11 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.17 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

- After this, now its time to loading data using **LoadData.sql** file

```
mysql> \. LoadData.sql
Query OK, 974 rows affected, 974 warnings (0.22 sec)
Records: 974 Deleted: 0 Skipped: 0 Warnings: 974

Query OK, 11113 rows affected, 11346 warnings (0.97 sec)
Records: 11113 Deleted: 0 Skipped: 0 Warnings: 11346

Query OK, 1698 rows affected, 3 warnings (0.31 sec)
Records: 1698 Deleted: 0 Skipped: 0 Warnings: 3

Query OK, 92 rows affected, 92 warnings (0.15 sec)
Records: 92 Deleted: 0 Skipped: 0 Warnings: 92

Query OK, 329 rows affected, 329 warnings (0.17 sec)
Records: 329 Deleted: 0 Skipped: 0 Warnings: 329

Query OK, 1555 rows affected, 7700 warnings (0.56 sec)
Records: 2315 Deleted: 0 Skipped: 700 Warnings: 7700

Query OK, 3895 rows affected, 3897 warnings (0.65 sec)
Records: 3895 Deleted: 0 Skipped: 0 Warnings: 3897

mysql>
```

- In this step, we will try some queries using **Queries.sql** file. As **Queries.sql** file contains almost 20 queries. If I source the file it will show me all result together. So here, I am trying to attempt every query one to one.

This shows me the **ATHLETE TABLE** description. I have 7 tables like this.

```
mysql> DESC ATHLETE ;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| athlete_code | varchar(100) | NO | PRI | NULL | |
| name | char(70) | NO | | NULL | |
| name_short | varchar(100) | NO | | NULL | |
| name_tv | varchar(90) | NO | | NULL | |
| gender | char(1) | NO | | NULL | |
| country_code | char(3) | NO | | NULL | |
| country_long | varchar(100) | NO | | NULL | |
| disciplines | varchar(200) | NO | | NULL | |
| events | varchar(100) | NO | | NULL | |
+-----+
```

7. This one shows the **TOTAL_MEDAL TABLE** Description.

```
mysql> DESC TOTAL_MEDAL;
```

Field	Type	Null	Key	Default	Extra
country_code	char(3)	NO		NULL	
country_long	varchar(100)	NO		NULL	
Gold_Medal	int	YES		NULL	
Silver_Medal	int	YES		NULL	
Bronze_Medal	int	YES		NULL	
Total	int	YES		NULL	
medal_id	int	NO	PRI	NULL	auto_increment

8. Now I am trying to retrieve all the data from **TOTAL_MEDAL TABLE** Like this:

```
mysql> SELECT * FROM TOTAL_MEDAL;
```

country_code	country_long	Gold_Medal	Silver_Medal	Bronze_Medal	Total	medal_id
USA	United States of America	40	44	42	126	1
CHN	People's Republic of China	40	27	24	91	2
JPN	Japan	20	12	13	45	3
AUS	Australia	18	19	16	53	4
FRA	France	16	26	22	64	5
NED	Netherlands	15	7	12	34	6
GBR	Great Britain	14	22	29	65	7
KOR	Republic of Korea	13	9	10	32	8
ITA	Italy	12	13	15	40	9
GER	Germany	12	13	8	33	10
NZL	New Zealand	10	7	3	20	11
CAN	Canada	9	7	11	27	12
UZB	Uzbekistan	8	2	3	13	13
HUN	Hungary	6	7	6	19	14
ESP	Spain	5	4	9	18	15
SWE	Sweden	4	4	3	11	16
KEN	Kenya	4	2	5	11	17
NOR	Norway	4	1	3	8	18
IRL	Ireland	4	0	3	7	19
BRA	Brazil	3	7	10	20	20
IRI	Islamic Republic of Iran	3	6	3	12	21
UKR	Ukraine	3	5	4	12	22
ROU	Romania	3	4	2	9	23
GEO	Georgia	3	3	1	7	24
BLR	Belarus	3	2	2	7	25

It has 92 Rows .

Now I am trying to retrieve all the data from **EVENT TABLE** Like this:

```
mysql> SELECT * FROM EVENT;
```

event	sport	sport_code	event_id
Men's Individual	Archery	ARC	1
Women's Individual	Archery	ARC	2
Men's Team	Archery	ARC	3
Women's Team	Archery	ARC	4
Mixed Team	Archery	ARC	5
Men's Team	Artistic Gymnastics	GAR	6
Men's All-Around	Artistic Gymnastics	GAR	7
Men's Floor Exercise	Artistic Gymnastics	GAR	8
Men's Pommel Horse	Artistic Gymnastics	GAR	9
Men's Rings	Artistic Gymnastics	GAR	10
Men's Vault	Artistic Gymnastics	GAR	11
Men's Parallel Bars	Artistic Gymnastics	GAR	12
Men's Horizontal Bar	Artistic Gymnastics	GAR	13
Women's Team	Artistic Gymnastics	GAR	14
Women's All-Around	Artistic Gymnastics	GAR	15
Women's Vault	Artistic Gymnastics	GAR	16
Women's Uneven Bars	Artistic Gymnastics	GAR	17
Women's Balance Beam	Artistic Gymnastics	GAR	18
Women's Floor Exercise	Artistic Gymnastics	GAR	19
Duet	Artistic Swimming	SWA	20
Team	Artistic Swimming	SWA	21
Men's 100m	Athletics	ATH	22
Men's 200m	Athletics	ATH	23
Men's 400m	Athletics	ATH	24
Men's 800m	Athletics	ATH	25
Men's 1500m	Athletics	ATH	26
Men's 5000m	Athletics	ATH	27
Men's 10000m	Athletics	ATH	28
Men's 20000m	Athletics	ATH	29
Men's 40000m	Athletics	ATH	30
Men's 80000m	Athletics	ATH	31
Men's 160000m	Athletics	ATH	32
Men's 320000m	Athletics	ATH	33
Men's 640000m	Athletics	ATH	34
Men's 1280000m	Athletics	ATH	35
Men's 2560000m	Athletics	ATH	36
Men's 5120000m	Athletics	ATH	37
Men's 10240000m	Athletics	ATH	38
Men's 20480000m	Athletics	ATH	39
Men's 40960000m	Athletics	ATH	40
Men's 81920000m	Athletics	ATH	41
Men's 163840000m	Athletics	ATH	42
Men's 327680000m	Athletics	ATH	43
Men's 655360000m	Athletics	ATH	44
Men's 1310720000m	Athletics	ATH	45
Men's 2621440000m	Athletics	ATH	46
Men's 5242880000m	Athletics	ATH	47
Men's 10485760000m	Athletics	ATH	48
Men's 20971520000m	Athletics	ATH	49
Men's 41943040000m	Athletics	ATH	50
Men's 83886080000m	Athletics	ATH	51
Men's 167772160000m	Athletics	ATH	52
Men's 335544320000m	Athletics	ATH	53
Men's 671088640000m	Athletics	ATH	54
Men's 1342177280000m	Athletics	ATH	55
Men's 2684354560000m	Athletics	ATH	56
Men's 5368709120000m	Athletics	ATH	57
Men's 10737418240000m	Athletics	ATH	58
Men's 21474836480000m	Athletics	ATH	59
Men's 42949672960000m	Athletics	ATH	60
Men's 85899345920000m	Athletics	ATH	61
Men's 171798691840000m	Athletics	ATH	62
Men's 343597383680000m	Athletics	ATH	63
Men's 687194767360000m	Athletics	ATH	64
Men's 1374389534720000m	Athletics	ATH	65
Men's 2748779069440000m	Athletics	ATH	66
Men's 5497558138880000m	Athletics	ATH	67
Men's 10995116277760000m	Athletics	ATH	68
Men's 21990232555520000m	Athletics	ATH	69
Men's 43980465111040000m	Athletics	ATH	70
Men's 87960930222080000m	Athletics	ATH	71
Men's 175921860444160000m	Athletics	ATH	72
Men's 351843720888320000m	Athletics	ATH	73
Men's 703687441776640000m	Athletics	ATH	74
Men's 1407374883553280000m	Athletics	ATH	75
Men's 2814749767106560000m	Athletics	ATH	76
Men's 5629499534213120000m	Athletics	ATH	77
Men's 11258999068426240000m	Athletics	ATH	78
Men's 22517998136852480000m	Athletics	ATH	79
Men's 45035996273704960000m	Athletics	ATH	80
Men's 90071992547409920000m	Athletics	ATH	81
Men's 180143985094819840000m	Athletics	ATH	82
Men's 360287970189639680000m	Athletics	ATH	83
Men's 720575940379279360000m	Athletics	ATH	84
Men's 1441151880758558720000m	Athletics	ATH	85
Men's 2882303761517117440000m	Athletics	ATH	86
Men's 5764607523034234880000m	Athletics	ATH	87
Men's 11529215046068469760000m	Athletics	ATH	88
Men's 23058430092136939520000m	Athletics	ATH	89
Men's 46116860184273879040000m	Athletics	ATH	90
Men's 92233720368547758080000m	Athletics	ATH	91
Men's 184467440737095516160000m	Athletics	ATH	92
Men's 368934881474191032320000m	Athletics	ATH	93
Men's 737869762948382064640000m	Athletics	ATH	94
Men's 1475739525896764129280000m	Athletics	ATH	95
Men's 2951479051793528258560000m	Athletics	ATH	96
Men's 5902958103587056517120000m	Athletics	ATH	97
Men's 11805916207174113034240000m	Athletics	ATH	98
Men's 23611832414348226068480000m	Athletics	ATH	99
Men's 47223664828696452136960000m	Athletics	ATH	100
Men's 94447329657392904273920000m	Athletics	ATH	101
Men's 188894659314785808547840000m	Athletics	ATH	102
Men's 377789318629571617095680000m	Athletics	ATH	103
Men's 755578637259143234191360000m	Athletics	ATH	104
Men's 1511157274518286468382720000m	Athletics	ATH	105
Men's 3022314549036572936765440000m	Athletics	ATH	106
Men's 6044629098073145873530880000m	Athletics	ATH	107
Men's 12089258196146291747061760000m	Athletics	ATH	108
Men's 24178516392292583494123520000m	Athletics	ATH	109
Men's 48357032784585166988247040000m	Athletics	ATH	110
Men's 96714065569170333976494080000m	Athletics	ATH	111
Men's 193428131138340667952988160000m	Athletics	ATH	112
Men's 386856262276681335905976320000m	Athletics	ATH	113
Men's 773712524553362671811952640000m	Athletics	ATH	114
Men's 1547425049106725343623905280000m	Athletics	ATH	115
Men's 3094850098213450687247810560000m	Athletics	ATH	116
Men's 6189700196426901374495621120000m	Athletics	ATH	117
Men's 12379400392853802748991242240000m	Athletics	ATH	118
Men's 24758800785707605497982484480000m	Athletics	ATH	119
Men's 49517601571415210995964968960000m	Athletics	ATH	120
Men's 99035203142830421991929937920000m	Athletics	ATH	121
Men's 198070406285660843983859875840000m	Athletics	ATH	122
Men's 396140812571321687967719751680000m	Athletics	ATH	123
Men's 792281625142643375935439503360000m	Athletics	ATH	124
Men's 1584563250285286751870879006720000m	Athletics	ATH	125
Men's 3169126500570573503741758013440000m	Athletics	ATH	126
Men's 6338253001141147007483516026880000m	Athletics	ATH	127
Men's 12676506002282294014967032053760000m	Athletics	ATH	128
Men's 25353012004564588029934064107520000m	Athletics	ATH	129
Men's 50706024009129176059868128215040000m	Athletics	ATH	130
Men's 101412048018258352119736256430080000m	Athletics	ATH	131
Men's 202824096036516704239472512860160000m	Athletics	ATH	132
Men's 405648192073033408478945025720320000m	Athletics	ATH	133
Men's 811296384146066816957890051440640000m	Athletics	ATH	134
Men's 1622592768292133633915780102881280000m	Athletics	ATH	135
Men's 3245185536584267267831560205762560000m	Athletics	ATH	136
Men's 6490371073168534535663120411525120000m	Athletics	ATH	137
Men's 12980742146337069071326240823050240000m	Athletics	ATH	138
Men's 25961484292674138142652481646100480000m	Athletics	ATH	139
Men's 51922968585348276285304963292200960000m	Athletics	ATH	140
Men's 103845937170696552570609926584401920000m	Athletics	ATH	141
Men's 207691874341393105141219853168803840000m	Athletics	ATH	142
Men's 415383748682786210282439706337607680000m	Athletics	ATH	143
Men's 830767497365572420564879412675215360000m	Athletics	ATH	144
Men's 1661534994731144841129758825350430720000m	Athletics	ATH	145
Men's 3323069989462289682259517650700861440000m	Athletics	ATH	146
Men's 6646139978924579364519035301401722880000m	Athletics	ATH	147
Men's 13292279957849158729038070602803445760000m	Athletics	ATH	148
Men's 26584559915698317458076141205606891520000m	Athletics	ATH	149
Men's 53169119831396634916152282411213783040000m	Athletics	ATH	150
Men's 106338239662793269832304564822427566080000m	Athletics	ATH	151
Men's 212676479325586539664609129644855132160000m	Athletics	ATH	152
Men's 425352958651173079329218259289710264320000m	Athletics	ATH	153
Men's 850705917302346158658436518579420528640000m	Athletics	ATH	154
Men's 1701411834604692317316873037158841057280000m	Athletics	ATH	155
Men's 3402823669209384634633746074317682114560000m	Athletics	ATH	156
Men's 6805647338418769269267492148635364229120000m	Athletics	ATH	157
Men's 13611294676837538538534984297270728458240000m	Athletics	ATH	158
Men's 27222589353675077077069968594541456916480000m	Athletics	ATH	159
Men's 54445178707350154154139937189082913832960000m	Athletics	ATH	160
Men's 108890357414700308308279874378165827665920000m	Athletics	ATH	161
Men's 217780714829400616616559748756331655331840000m	Athletics	ATH	162
Men's 435561429658801233233119497512663310663680000m	Athletics	ATH	163
Men's 871122859317602466466238995025326621327360000m	Athletics	ATH	164
Men's 1742245718635204932932477990050653242654720000m	Athletics	ATH	165
Men's 3484491437270409865864955980101306485309440000m	Athletics	ATH	166
Men's 6968982874540819731729911960202612970618880000m	Athletics	ATH	167
Men's 13937965749081639463459823920405225941237760000m	Athletics	ATH	168
Men's 27875931498163278926919647840810451882475520000m	Athletics	ATH	169
Men's 55751862996326557853839295681620903764951040000m	Athletics	ATH	170
Men's 111503725992653115707678591363241807529902080000m	Athletics	ATH	171
Men's 223007451985306231415357182726483615059804160000m	Athletics	ATH	172
Men's 446014903970612462830714365452967230119608320000m	Athletics	ATH	173
Men's 892029807941224925661428730905934460239216640000m	Athletics	ATH	174
Men's 1784059615882449851322857461811868920478433280000m	Athletics	ATH	175
Men's 3568119231764899702645714923623737840956866560000m	Athletics	ATH	176
Men's 7136238463529799405291429847247475681913733120000m	Athletics	ATH	177
Men's 14272476927059598810582859694494951363827466240000m	Athletics	ATH	178
Men's 285449538541191			


```
mysql> SELECT * FROM COACH;
```

coach_code	coach_name	gender	coach_function	country_code	country_long	disciplines	events
1533246	PEDRERO Ofelia	Fem	Coach	MEX	Mexico	Artistic Swimming	Team
1533775	RAHMTI SHENAZISHIL	Mal	Head Coach	IRQ	Iraq	Football	Men
1536655	AFLAKHMANSEH Hajid	Mal	Coach	IRI	Islamic Republic of Iran	Taekwondo	
1536659	YOUSSEFY Mehرداد	Mal	Coach	IRI	Islamic Republic of Iran	Taekwondo	
1536668	MAZOUH Haseo	Fem	Coach	IRI	Islamic Republic of Iran	Taekwondo	
1536326	LOFTUS Adriane	Fem	Coach	MEX	Mexico	Artistic Swimming	Team
1538313	FERRARA Fernando	Mal	Head Coach	ARG	Argentina	Hockey	
1538315	GULLA Alejandra	Fem	Assistant Coach	ARG	Argentina	Hockey	Women
1538317	CAPURRO Santiago	Mal	Assistant Coach	ARG	Argentina	Hockey	
1538745	RONCONI Mariano	Mal	Head Coach	ARG	Argentina	Hockey	
1538748	PAULON Ezequiel	Mal	Assistant Coach	ARG	Argentina	Hockey	
1538751	VILA Matias	Mal	Assistant Coach	ARG	Argentina	Hockey	
1539586	RHEZKHAH HAJIRASOULIAN Mohammadre.	Mal	Coach	IRI	Islamic Republic of Iran	Artistic Gymnastics	
1548258	DAVIS DIAZ David	Mal	Coach	MEX	Mexico	Taekwondo	
1548259	MENDOZA MONA Abel	Mal	Coach	MEX	Mexico	Taekwondo	
1548260	VICTORIA ESPINOSA de los Alfonso	Mal	Coach	MEX	Mexico	Taekwondo	
1548622	NIJLAND Guillermo	Mal	Head Coach	ARG	Argentina	Handball	Men
1548636	GOMEZ CORA Santiago	Mal	Head Coach	ARG	Argentina	Rugby Sevens	Men
1548639	CRAVANO Leonardo	Mal	Assistant Coach	ARG	Argentina	Rugby Sevens	Men
1548840	FILTORIANU Ana Luliza	Fem	Coach	ROU	Romania	Rhythmic Gymnastics	

10. Retrieving data from TEAM TABLE

```
mysql> SELECT * FROM TEAM;
```

code	team_gender	country_code	country	country_long	discipline	disciplines_code	events	non_athletes
ARCHTEAMS---CHN1	M	CHN	China	People's Republic of China	Archery	ARC	Men's Team	3
ARCHTEAMS---COL1	M	COL	Colombia	Colombia	Archery	ARC	Men's Team	3
ARCHTEAMS---FRA1	M	FRA	France	France	Archery	ARC	Men's Team	3
ARCHTEAMS---GBR1	M	GBR	Great Britain	Great Britain	Archery	ARC	Men's Team	3
ARCHTEAMS---IND1	M	IND	India	India	Archery	ARC	Men's Team	3
ARCHTEAMS---ITA1	M	ITA	Italy	Italy	Archery	ARC	Men's Team	3
ARCHTEAMS---JPN1	M	JPN	Japan	Japan	Archery	ARC	Men's Team	3
ARCHTEAMS---KAZ1	M	KAZ	Kazakhstan	Kazakhstan	Archery	ARC	Men's Team	3
ARCHTEAMS---KOR1	M	KOR	Korea	Republic of Korea	Archery	ARC	Men's Team	3
ARCHTEAMS---MEX1	M	MEX	Mexico	Mexico	Archery	ARC	Men's Team	3
ARCHTEAMS---TPE1	M	TPE	Chinese Taipei	Chinese Taipei	Archery	ARC	Men's Team	3
ARCHTEAMS---TUR1	M	TUR	Turkiye	Turkiye	Archery	ARC	Men's Team	3
ARCHTEAMS---CHN1	W	CHN	China	People's Republic of China	Archery	ARC	Women's Team	3
ARCHTEAMS---FRA1	W	FRA	France	France	Archery	ARC	Women's Team	3
ARCHTEAMS---GBR1	W	GBR	Great Britain	Great Britain	Archery	ARC	Women's Team	3
ARCHTEAMS---GER1	W	GER	Germany	Germany	Archery	ARC	Women's Team	3
ARCHTEAMS---INA1	W	INA	Indonesia	Indonesia	Archery	ARC	Women's Team	3
ARCHTEAMS---IND1	W	IND	India	India	Archery	ARC	Women's Team	3
ARCHTEAMS---KOR1	W	KOR	Korea	Republic of Korea	Archery	ARC	Women's Team	3
ARCHTEAMS---MAS1	W	MAS	Malaysia	Malaysia	Archery	ARC	Women's Team	3
ARCHTEAMS---MEX1	W	MEX	Mexico	Mexico	Archery	ARC	Women's Team	3
ARCHTEAMS---NED1	W	NED	Netherlands	Netherlands	Archery	ARC	Women's Team	3
ARCHTEAMS---TPE1	W	TPE	Chinese Taipei	Chinese Taipei	Archery	ARC	Women's Team	3
ARCHTEAMS---USA1	W	USA	United States	United States of America	Archery	ARC	Women's Team	3
ARCHTEAMS---AUS1	X	AUS	Australia	Australia	Archery	ARC	Mixed Team	2
ARCHTEAMS---BRA1	X	BRA	Brazil	Brazil	Archery	ARC	Mixed Team	2
ARCHTEAMS---CAN1	X	CAN	Canada	Canada	Archery	ARC	Mixed Team	2
ARCHTEAMS---CHN1	X	CHN	China	People's Republic of China	Archery	ARC	Mixed Team	2
ARCHTEAMS---COL1	X	COL	Colombia	Colombia	Archery	ARC	Mixed Team	2
ARCHTEAMS---CZE1	X	CZE	Czechia	Czechia	Archery	ARC	Mixed Team	2
ARCHTEAMS---EGY1	X	EGY	Egypt	Egypt	Archery	ARC	Mixed Team	2

It has rows of 1698.

5. Implement Queries

- By this queries we can get the discipline name which has occur venue name ‘**champ-de-Mars arena**’

```
mysql> SELECT
->     DISTINCT discipline
-> FROM
->     SCHEDULE
-> WHERE
->     venue = 'Champ-de-Mars Arena';

+-----+
| discipline |
+-----+
| Judo       |
| Wrestling  |
+-----+
2 rows in set (0.01 sec)
```

2. By this query we can get out the total athletes number and renamed the table as **TOTAL_ATHLETES FROM TEAM TABLE.**

```
mysql> SELECT SUM(num_athletes) AS TOTAL_ATHLETES FROM TEAM;
+-----+
| TOTAL_ATHLETES |
+-----+
|          7808 |
+-----+
1 row in set (0.00 sec)
```

- This query help us to get all the team code from **TEAM TABLE** where discipline is '%Archery%';

```
mysql> SELECT code AS TEAM_code FROM TEAM where discipline like '%Archery%';
+-----+
| TEAM_code |
+-----+
| ARCHTEAM3---CHN01 |
| ARCHTEAM3---COL01 |
| ARCHTEAM3---FRA01 |
| ARCHTEAM3---GBR01 |
| ARCHTEAM3---IND01 |
| ARCHTEAM3---ITA01 |
| ARCHTEAM3---JPN01 |
| ARCHTEAM3---KAZ01 |
| ARCHTEAM3---KOR01 |
| ARCHTEAM3---MEX01 |
| ARCHTEAM3---TPE01 |
| ARCHTEAM3---TUR01 |
| ARCHTEAM3---CHN01 |
| ARCHTEAM3---FRA01 |
| ARCHTEAM3---GBR01 |
| ARCHTEAM3---GER01 |
| ARCHTEAM3---INA01 |
| ARCHTEAM3---IND01 |
| ARCHTEAM3---KOR01 |
| ARCHTEAM3---MAS01 |
| ARCHTEAM3---MEX01 |
```

It display 57 rows.

4. From **ATHLETE TABLE** get out the country name, count athlete based on the country. renamed the table as Total participant

```
mysql> SELECT
->   country_long AS Name_of_Country,
->   COUNT(athlete_code) AS Total_Participants
-> FROM
->   ATHLETE
-> GROUP BY
->   country_long
-> ORDER BY
->   country_long ;
```

Name_of_Country	Total_Participants
Afghanistan	0
AIN	31
Albania	8
Algeria	46
American Samoa	2
Andorra	2
Angola	25
Antigua and Barbuda	5
Argentina	143
Armenia	15
Aruba	6
Australia	475
Austria	84
Azerbaijan	46
Bahamas	19
Bahrain	14
Bangladesh	5
Barbados	5

it displays 206 Rows

- From **COACH TABLE** get out the country name, count coach based on the country. renamed the table as **Total coches num by country**

```
mysql> SELECT
->   country_long,
->   COUNT(coach_name) AS Total_Coaches_NUM_BY_Country
-> FROM
->   COACH
-> GROUP BY
->   country_long
-> ORDER BY
->   country_long;
```

country_long	Total_Coaches_NUM_BY_Country
AIN	3
Algeria	3
Angola	3
Argentina	19
Armenia	2
Australia	40
Austria	3
Azerbaijan	6
Belgium	17
Brazil	35
Bulgaria	9
Burkina Faso	3
Canada	34
Chile	1
Chinese Taipei	3
Colombia	5
Côte d'Ivoire	2
Croatia	10

diaplays around 98 rows.

6. This query to get out the maximum event name from **TEAM** using aggregation function

```
mysql> Select MAX(events) AS EVENT_MAX from TEAM
-> ;
+-----+
| EVENT_MAX |
+-----+
| Women's Team Sprint |
+-----+
1 row in set (0.00 sec)
```

- ## 7. Athletes and coach ratio from **ATHLTES AND CAOCH TABLE**

```
mysql> SELECT
->   c.country_code,
->   COUNT(DISTINCT a.athlete_code) AS Total_athletes,
->   COUNT(DISTINCT c.coach_code) AS Total_coaches,
->   ROUND(COUNT(DISTINCT a.athlete_code) / COUNT(DISTINCT c.coach_code), 2) AS Athletes_vs_coach_Ratio
-> FROM
->   COACH c
-> 2021
->   ATHLETE a ON c.country_code = a.country_code      -- Joining based on country_code
-> GROUP BY
->   c.country_code
-> ORDER BY
->   Total_athletes DESC , Total_coaches DESC;
```

country_code	Total_athletes	Total_coaches	Athletes_vs_coach_Ratio
USA	619	56	11.05
FRA	601	55	10.93
AUS	475	48	11.88
GER	437	37	12.35
JPN	431	43	10.02
ESP	401	68	5.90
CHN	386	45	8.64
ITA	397	23	17.26
GBR	343	31	10.72
CAN	332	34	9.76
RUS	295	35	8.29
NED	290	28	14.59
POL	226	7	32.29
BLR	208	22	9.45

it displays 98 rows.

8. List All Coaches Who Have Coached Athletes in More Than 30 Discipline

[illegible]

it displays 488 rows

9. Get the most medals won by each country, using `limit` get the top 10


```
mysql> select country, lang as Country_Won_Most_medals, Total
-> from TOTAL_MEDAL
-> order by
-> Total DESC
-> limit 10;
```

Country_Won_Most_medals	Total
United States of America	126
People's Republic of China	91
Great Britain	85
France	54
Australia	53
Japan	45
Italy	40
Netherlands	34
Germany	33
Republic of Korea	32

10 rows in set (0.00 sec)

10. get total_events, total_athletes, total_country and total medals select queries.

```
mysql> SELECT
-> (SELECT COUNT(DISTINCT event_id) FROM EVENT) AS Total_Events,
-> (SELECT COUNT(DISTINCT country_code) FROM ATHLETE) AS Total_Countries,
-> (SELECT COUNT(DISTINCT athlete_code) FROM ATHLETE) AS Total_Athletes,
-> (SELECT SUM(Gold_Medal + Silver_Medal + Bronze_Medal) FROM TOTAL_MEDAL) AS Total_Medals;
```

Total_Events	Total_Countries	Total_Athletes	Total_Medals
329	200	11113	1043

1 row in set (0.03 sec)

11. calculates the male participants percentages in each discipline Using **ATHLETE table**

```
mysql> SELECT
-> a.disciplines AS Games_participated_by_male,
-> SUM(a.gender = 'M') AS Male, -- counts male participants using sum aggregation function
-> COUNT(a.athlete_code) AS Total_athlete_num, -- Total number of participants
-> ROUND((SUM(a.gender = 'M') / COUNT(a.athlete_code)) * 100, 2) AS Male_Participation -- Male participation percentage
-> FROM
-> ATHLETE a
-> GROUP BY
-> a.disciplines
-> ORDER BY
-> Male_Participation DESC;
```

Games_participated_by_male	Male	Total_athlete_num	Male_Participation
[Athletics]	2	2	100.00
[Swimming]	1	1	100.00
[Marathon Swimming, 'Swimming']	13	17	76.47
[Wrestling]	195	201	97.01
[Boxing]	140	142	98.59
[Football]	313	313	99.99
[Water Polo]	150	150	100.00
[Swimming]	410	410	100.00
[Cycling Road]	90	114	78.94
[Tennis]	60	114	52.63
[Cycling Mountain Bike]	35	68	51.47
[Triathlon]	57	113	50.44
[Archery]	1000	1019	98.13
[Canoe Sprint]	41	84	48.81
[Hockey]	213	415	51.33
[Cycling Track]	114	224	50.89

it displays 52 rows.

12. calculates the Female participants percentages in each discipline Using **ATHLETE table**

```
mysql> SELECT
-> a.disciplines AS Games_participated_by_female,
-> SUM(a.gender = 'F') AS Female,
-> COUNT(a.athlete_code) AS Total_athlete_num,
-> ROUND((SUM(a.gender = 'F') / COUNT(a.athlete_code)) * 100, 2) AS Female_Participation
-> FROM
-> ATHLETE a
-> GROUP BY
-> a.disciplines
-> ORDER BY
-> Female_Participation DESC;
```

Games_participated_by_female	Female	Total_athlete_num	Female_Participation
[3x3 Basketball, 'Basketball']	1	1	100.00
[Cycling Road, 'Triathlon']	1	1	100.00
[Rhythmic Gymnastics]	94	94	100.00
[Artistic Swimming]	166	166	100.00
[Cycling Road, 'Cycling Track']	8	11	72.73
[Marathon Swimming]	20	30	66.67
[Handball]	202	306	65.99
[Breaking]	17	33	51.52
[Diving]	68	135	50.37
[Table Tennis]	88	175	50.29
[Canoe Sprint]	120	239	50.21
[Cycling BMX Freestyle]	12	24	50.00
[Trampoline Gymnastics]	16	32	50.00
[Golf]	60	120	50.00
[Skateboarding]	44	88	50.00
[Beach Volleyball]	48	96	50.00

it displays 52 rows.

13. get the coach name ASSOCIATED WITH COUNTRY AND won least 10 gold medals , used join and sub-query

```
mysql> SELECT
-> c.coach_name,
-> c.country_code AS COUNTRY,
-> c.disciplines AS GAMES
-> FROM
-> COACH c
-> INNER JOIN (
-> SELECT
-> country_code,
-> SUM(Gold_Medal) AS GOLD_TOTAL,
-> FROM
-> TOTAL_MEDAL
-> GROUP BY
-> country_code
-> HAVING
-> SUM(Gold_Medal) >= 10
-> ) AS WINNER_OF_MEDAL
-> ON c.country_code = WINNER_OF_MEDAL.country_code;
```

coach_name	COUNTRY	GAMES
THORPE Karen	GBR	Artistic Swimming
TOMOMATSU Yuniko	GBR	Artistic Swimming
BUSNARI Alberto	ITA	Artistic Gymnastics
COCCHIARDI Giuseppe	ITA	Artistic Gymnastics
FORTUNA Marco	ITA	Artistic Gymnastics
BERGAMELLI Monica Roberta	ITA	Artistic Gymnastics
ALTENBURG Valentin	GER	Hockey
HENNING Andre	GER	Hockey

it displays 388 rows.

- retrieve the current age of the athlete using DATEDIFF format and renamed the table CURRENT_AGE_ATHLETS, name, gender, medal type and discipline from MEDALLIST TABLE

```
mysql> SELECT
-> name,
-> birth_date,
-> FLOOR(DATEDIFF(CURDATE(), birth_date) / 365.25) AS CURRENT_AGE_ATHLETS,
-> gender,
-> medal_type,
-> discipline
-> FROM
-> MEDALLIST;
```

name	birth_date	CURRENT_AGE_ATHLETS	gender	medal_type	discipline
CHANG Yanl	2001-12-07	22	F	Gold Medal	Diving
CHEN Ylwen	1999-06-15	25	F	Gold Medal	Diving
BACON Sarah	1996-09-20	28	F	Silver Medal	Diving
COOK Cassidy	1995-05-09	29	F	Silver Medal	Diving
HASPER Yasin	2000-07-20	24	F	Bronze Medal	Diving
HEW JENSEN Scarlett	2001-12-31	22	F	Bronze Medal	Diving
PASQUET Varian	1999-07-29	25	M	Gold Medal	Rugby Sevens
TIMO Andy	2004-05-28	20	M	Gold Medal	Rugby Sevens
REBBADJ Rayan	1999-08-15	25	M	Gold Medal	Rugby Sevens
FORNER Theo	2001-10-17	23	M	Gold Medal	Rugby Sevens
PAREZ EDO MARTIN Stephen	1994-08-01	30	M	Gold Medal	Rugby Sevens
RIVA Paulin	1994-04-20	30	M	Gold Medal	Rugby Sevens
JOSEPH Jefferson-Lee	2002-08-29	22	M	Gold Medal	Rugby Sevens
ZECHDAR Antoine	1999-05-22	25	M	Gold Medal	Rugby Sevens
GRANDIDIER NKANANG Aaron	2000-05-18	24	M	Gold Medal	Rugby Sevens
BARRAQUE Jean Pascal	1991-04-24	33	M	Gold Medal	Rugby Sevens
DUPOUR Antoine	1996-11-15	27	M	Gold Medal	Rugby Sevens
SEPHO Jordan	1998-12-08	25	M	Gold Medal	Rugby Sevens

It

displays 1555 rows

- Compare medals between Female & Male

```
mysql> SELECT
-> gender,
-> discipline,
-> COUNT(*) AS total_medals,
-> SUM(CASE WHEN medal_type = 'Gold Medal' THEN 1 ELSE 0 END) AS Gold_Medals,
-> SUM(CASE WHEN medal_type = 'Silver Medal' THEN 1 ELSE 0 END) AS Silver_Medals,
-> SUM(CASE WHEN medal_type = 'Bronze Medal' THEN 1 ELSE 0 END) AS Bronze_Medals
-> FROM
-> MEDALLIST
-> GROUP BY
-> discipline, gender
-> ORDER BY
-> discipline;
```

gender	discipline	total_medals	Gold_Medals	Silver_Medals	Bronze_Medals
F	3x3 Basketball	12	4	4	4
M	3x3 Basketball	12	4	4	4
F	Archery	12	4	4	4
M	Archery	12	4	4	4
F	Artistic Gymnastics	15	5	5	5
M	Artistic Gymnastics	15	5	5	5
F	Artistic Swimming	33	11	11	11
F	Athletics	54	18	17	19
M	Athletics	50	18	15	17
F	Badminton	9	3	3	3
M	Badminton	9	3	3	3

it display 54 rows

- find out the Medals by discipline

```
mysql> SELECT discipline, COUNT(*) AS total_medals
-> FROM MEDALLIST
-> GROUP BY discipline
-> ORDER BY total_medals DESC;
```

discipline	total_medals
Rowing	138
Swimming	135
Football	124
Athletics	104
Hockey	102
Handball	94
Rugby Sevens	78
Water Polo	78
Volleyball	78
Basketball	72
Fencing	72
Cycling Track	69
Canoe Sprint	50
Judo	49
Equestrian	36
Artistic Swimming	33
Artistic Gymnastics	30
Diving	24
3x3 Basketball	24
Sailing	24
Table Tennis	24
Archery	24
Tennis	18
Badminton	18

shows 28 rows

17. list down the athletes age group and compare how many medals(medal_type) they have win based on age group

```
mysql> SELECT
-> CASE
-> WHEN YEAR(CURDATE()) - YEAR(birth_date) < 20 THEN 'UNDER 20'
-> WHEN YEAR(CURDATE()) - YEAR(birth_date) BETWEEN 20 AND 29 THEN 'AGE between 20 to 29'
-> WHEN YEAR(CURDATE()) - YEAR(birth_date) BETWEEN 30 AND 39 THEN 'AGE between 30 to 39'
-> ELSE 'AGE OVER 40'
-> END AS ATHLETES_AGE_GROUP,
-> gender,
-> COUNT(*) AS MEDAL_TOTAL,
-> SUM(CASE WHEN medal_type = 'Gold Medal' THEN 1 ELSE 0 END) AS Gold_Medals,
-> SUM(CASE WHEN medal_type = 'Silver Medal' THEN 1 ELSE 0 END) AS Silver_Medals,
-> SUM(CASE WHEN medal_type = 'Bronze Medal' THEN 1 ELSE 0 END) AS Bronze_Medals
-> FROM
-> MEDALLIST
-> GROUP BY
-> ATHLETES_AGE_GROUP, gender
-> ORDER BY
-> MEDAL_TOTAL DESC, ATHLETES_AGE_GROUP, gender;
```

ATHLETES_AGE_GROUP	gender	MEDAL_TOTAL	Gold_Medals	Silver_Medals	Bronze_Medals
AGE between 20 to 29	F	547	173	194	180
AGE between 20 to 29	M	502	171	158	173
AGE between 30 to 39	M	221	75	79	67
AGE between 30 to 39	F	207	78	60	69
UNDER 20	F	35	10	13	12
AGE OVER 40	M	19	4	6	9
UNDER 20	M	15	2	8	5
AGE OVER 40	F	9	4	2	3

8 rows in set (0.00 sec)

18. find out the youngest athlete using birthdate from MEDALLIST TABLE, also name, code of the athlete, gender, discipline, events, medal_type ,

```
mysql> SELECT
-> name ,
-> code_athlete,
-> gender,
-> birth_date,
-> discipline,
-> events,
-> medal_type,
-> medal_date
-> FROM
-> MEDALLIST
-> WHERE birth_date = (SELECT MIN(birth_date) FROM MEDALLIST);
```

name	code_athlete	gender	birth_date	discipline	events	medal_type	medal_date
KRAUT Laura	1951040	F	1965-11-14	Equestrian	Jumping Team	Silver Medal	2024-08-02

1 row in set (0.01 sec)

19. from MEDALLIST find out top Atheltes by counting medal , use limit to get only top 10

```
mysql> SELECT name, country_code, COUNT(*) AS medal_count
-> FROM MEDALLIST
-> GROUP BY name, country_code
-> ORDER BY medal_count DESC
-> LIMIT 10;
```

name	country_code	medal_count
O'CALLAGHAN Mollie	AUS	4
YANG Junxuan	CHN	4
HUSKE Torri	USA	3
McKEON Emma	AUS	3
WALSH Gretchen	USA	3
ZHANG Yufei	CHN	3
DRESSEL Caeleb	USA	3
JACK Shayna	AUS	2
SOUTHAM Flynn	AUS	2
HARRIS Meg	AUS	2

10 rows in set (0.00 sec)

20. from SCGEDULE TABLE get the start_date

```
mysql> SET time_zone = '+00:00';
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @@session.time_zone;
+-----+
| @@session.time_zone |
+-----+
| +00:00              |
+-----+
1 row in set (0.00 sec)

mysql> select start_date from SCHEDULE;
+-----+
| start_date |
+-----+
| 2024-07-24 13:00:00 |
| 2024-07-24 13:00:00 |
| 2024-07-24 13:30:00 |
| 2024-07-24 14:00:00 |
| 2024-07-24 14:30:00 |
| 2024-07-24 15:00:00 |
| 2024-07-24 15:00:00 |
| 2024-07-24 15:30:00 |
| 2024-07-24 16:00:00 |
| 2024-07-24 17:00:00 |
| 2024-07-24 17:00:00 |
| 2024-07-24 17:30:00 |
| 2024-07-24 18:00:00 |
| 2024-07-24 18:30:00 |
| 2024-07-24 19:00:00 |
| 2024-07-24 19:00:00 |
| 2024-07-24 19:00:00 |
```

returns 3895 rows displaying time in UTC

21. From SCHEDULE TABLE get the start_date, end_date, venue_code when status is cancelled

```
mysql> SELECT start_date, end_date, venue_code FROM SCHEDULE WHERE status = 'CANCELLED';
+-----+-----+-----+
| start_date | end_date | venue_code |
+-----+-----+-----+
| 2024-07-29 00:30:00 | 2024-07-29 00:30:00 | CPL |
| 2024-07-29 12:00:00 | 2024-07-29 12:00:00 | CPL |
| 2024-07-30 12:00:00 | 2024-07-30 12:00:00 | CPL |
| 2024-07-31 07:20:00 | 2024-07-31 07:20:00 | CPL |
| 2024-08-01 11:36:00 | 2024-08-01 11:57:00 | HAM |
| 2024-08-01 11:56:00 | 2024-08-01 12:17:00 | HAM |
| 2024-08-01 12:07:00 | 2024-08-01 12:28:00 | HAM |
| 2024-08-01 12:27:00 | 2024-08-01 12:48:00 | HAM |
| 2024-08-01 13:36:00 | 2024-08-01 13:59:00 | HAM |
| 2024-08-01 13:38:00 | 2024-08-01 13:59:00 | HAM |
| 2024-08-01 16:09:00 | 2024-08-01 16:30:00 | HAM |
| 2024-08-01 16:28:00 | 2024-08-01 16:49:00 | HAM |
| 2024-08-01 16:28:00 | 2024-08-01 16:49:00 | HAM |
| 2024-08-01 16:40:00 | 2024-08-01 17:01:00 | HAM |
| 2024-08-01 16:59:00 | 2024-08-01 17:20:00 | HAM |
| 2024-08-01 17:30:00 | 2024-08-01 17:51:00 | HAM |
| 2024-08-01 18:01:00 | 2024-08-01 18:22:00 | HAM |
| 2024-08-05 11:13:00 | 2024-08-05 11:14:00 | HAM |
| 2024-08-05 12:45:00 | 2024-08-05 13:30:00 | HAM |
| 2024-08-05 13:58:00 | 2024-08-05 14:41:00 | HAM |
| 2024-08-06 12:29:00 | 2024-08-06 13:19:00 | HAM |
| 2024-08-06 13:36:00 | 2024-08-06 14:26:00 | HAM |
| 2024-08-07 10:03:00 | 2024-08-07 10:20:00 | HAM |
| 2024-08-07 10:23:00 | 2024-08-07 10:40:00 | HAM |
| 2024-08-07 10:43:00 | 2024-08-07 11:00:00 | HAM |
| 2024-08-07 10:46:00 | 2024-08-07 11:05:00 | HAM |
| 2024-08-07 11:13:00 | 2024-08-07 11:30:00 | HAM |
| 2024-08-07 11:23:00 | 2024-08-07 11:40:00 | HAM |
| 2024-08-07 11:36:00 | 2024-08-07 11:55:00 | HAM |
| 2024-08-07 12:03:00 | 2024-08-07 12:20:00 | HAM |
```

it displays 70 rows

However, by sourcing **Query.sql** we can implement all these queries at a time and get all the result in a output file named **"SimpleQueryAnswerOlympic.out"**

6. Implementation of Advanced Features (Procedures, Views)

To implement the advance features of queries by using Procedure like user variable, if-else, cursor, store procedure In, out parameter one can source the file name “**advanceQuery.sql** “ which also have the output file named “**advanceQueryAnswerOlympic.out** “. now we will do the queries one by one.

Procedure _TASK1

store procedure to calculate the total medals on a given game and gender .

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE CAL_TOTAL_MEDALS(
  -- IN PRO_discipline VARCHAR(200),
  -- IN PRO_gender CHAR(1),
  -- OUT TotalMedals INT -- this will return the total medal number
  -- )
  -- COMMENT 'calucation of total medla fro a dicipline and gender'
  -- BEGIN
  -- -- declaring variable (Existsdiscipline,TotalCount )
  -- DECLARE Existsdiscipline INT DEFAULT 0; -- checks if discipline exists or not
  -- DECLARE TotalCount INT DEFAULT 0; -- this var hold the total count
  --
  -- SET TotalMedals = 0;
  --
  -- -- checkings the disclines existence
  -- SELECT COUNT(*) INTO Existsdiscipline
  -- FROM MEDALLIST
  -- WHERE discipline = PRO_discipline;
  --
  -- -- discipline does not exit, so display msg
  -- IF Existsdiscipline = 0 THEN
  -- -- SIGNAL SQLSTATE '45000' SET MESSAGE TEXT = 'Discipline does not exist.';
  -- -- SELECT CONCAT('Sorry!!! Discipline you have entered :', PRO_discipline, ' does not exist.') AS Total_Medal_CALCUALTION ;
  -- ELSE
  --
  -- -- calculates total medal number
  -- SELECT COUNT(*) INTO TotalCount
  -- FROM MEDALLIST
  -- WHERE discipline = PRO_discipline AND gender = PRO_gender;
  --
  -- -- now, the result(TotalCount) are assigning to out parameter which is TotalMedals
  -- SET TotalMedals = TotalCount;
  --
  -- END IF;
  -- END //
Query OK, 0 rows affected (0.02 sec)
```

1.Then calling the procedure for total of Male gender in Football and displaying the total using user variable

```
mysql> CALL CAL_TOTAL_MEDALS('Football', 'M', @total);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @total AS TOTAL_MEDALS;
+-----+
| TOTAL_MEDALS |
+-----+
|          62 |
+-----+
1 row in set (0.00 sec)
```

2.Now, calling a PROCEDURE for a discipline that does not exit and return 0 as a total medal

```
mysql> CALL CAL_TOTAL_MEDALS('Ludo', 'M', @total);
+-----+
| Total_Medal_CALCUALTION |
+-----+
| Sorry!!! Discipline you have entered :Ludo does not exist. |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @total AS TOTAL_MEDALS;
+-----+
| TOTAL_MEDALS |
+-----+
| 0 |
+-----+
1 row in set (0.00 sec)
```

3.For “water Polo” discipline get out the total medals in male gender

```
mysql> CALL CAL_TOTAL_MEDALS('Water Polo', 'M', @total);
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> SELECT @total AS TOTAL_MEDALS;
+-----+
| TOTAL_MEDALS |
+-----+
| 39 |
+-----+
1 row in set (0.00 sec)
```

Procedure _TASK2:

using procedure user variable that counting total medals that have been won in a discipline

```
mysql> DROP PROCEDURE IF EXISTS Counting_Medal_By_discipline;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE Counting_Medal_By_discipline(
-> IN discipline_name VARCHAR(200), -- input for discipline name
-> OUT medal_count INT -- out parametr for counting medals
-> )
->
-> COMMENT ' COUNTING total medal won on a specific discipline .'
-> BEGIN
-> SELECT COUNT(*) INTO medal_count
-> FROM MEDALLIST
-> WHERE discipline = discipline_name;
-> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL Counting_Medal_By_discipline('Handball', @totalMedals);
Query OK, 1 row affected (0.01 sec)
```

1. Calling procedure for handball to get the total medals for ti

```
mysql> CALL Counting_Medal_By_discipline('Handball', @totalMedals);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT @totalMedals AS Total_Medals;
+-----+
| Total_Medals |
+-----+
| 94 |
+-----+
1 row in set (0.00 sec)
```

2. Calling procedure to get the total medals for table tennis


```
mysql> CALL Counting_Medal_By_discipline('Table Tennis', @totalMedals);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @totalMedals AS Total_Medals;
+-----+
| Total_Medals |
+-----+
|          24 |
+-----+
1 row in set (0.00 sec)
```

Procedure _TASK3 :

store procedure using Cursor and LOOP to count medals (gold medals, silver medals, bronze medals) on a discipline

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE Count_Discipline(
-- IN discipline_name VARCHAR(200),
-- OUT count_gold INT,           -- out parameter for gold medal counting
-- OUT count_silver INT,        -- out parameter for silver medal counting
-- OUT count_bronze INT
-- )
-- BEGIN
-- -- declarign var (var_medal_type, done)
-- DECLARE var_medal_type VARCHAR(20);
-- DECLARE done INT DEFAULT 0; -- To control loop exit
--
-- -- declarign the cursor
-- DECLARE cursor_medal CURSOR FOR
-- SELECT medal_type
-- FROM MEDALLIST
-- WHERE discipline = discipline_name;
--
--
-- -- declar not found handler
-- DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
--
-- /*
-- /* -- initializing counts
-- /* SET count_gold = 0;
-- /* SET count_silver = 0;
-- /* SET count_bronze = 0;
-- /*
-- /*
-- -- open cursor(cursor_medal)
-- -- OPEN cursor_medal;
--
--
-- -- start loop
-- StartLoop: LOOP
--
-- -- fetch the cursor(cursor_medal) into variable( var_medal_type)
--
-- FETCH cursor_medal INTO var_medal_type;
--
--
-- -- checks cursor is done
-- IF done = 1 THEN
-- LEAVE StartLoop;
-- END IF;
--
--
-- -- checks the medal type by using if else
-- IF var_medal_type = 'Gold' THEN
-- SET count_gold = count_gold + 1;
-- ELSEIF var_medal_type = 'Silver' THEN
-- SET count_silver = count_silver + 1;
-- ELSEIF var_medal_type = 'Bronze' THEN
-- SET count_bronze = count_bronze + 1;
-- END IF;
--
-- END LOOP;
--
-- -- closing cursor
-- -- CLOSE cursor_medal;
-- -- END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
```

1. Calling the procedure for Water Polo to count the Gold Medal, silver medal, bronze medal

```
mysql> CALL Count_Discipline ('Water Polo', @count_gold, @count_silver, @count_bronze);
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @count_gold AS Gold_Medals, @count_silver AS Silver_Medals, @count_bronze AS Bronze_Medals;
+-----+-----+-----+
| Gold_Medals | Silver_Medals | Bronze_Medals |
+-----+-----+-----+
|          NULL |          NULL |          NULL |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Some reason, it is not showing me the proper value. Showing me NULL.

2.To DELETE all the procedure we can use this command like this

```
mysql> DROP PROCEDURE IF EXISTS CAL_TOTAL_MEDALS;
Query OK, 0 rows affected (0.03 sec)

mysql> DROP PROCEDURE IF EXISTS Counting_Medal_By_discipline;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

3. SHOW PROCEDURE STATUS WHERE Db='Olympic_Game_2024_22663281';

```
mysql> SHOW PROCEDURE STATUS WHERE Db='Olympic_Game_2024_22663281'; -- display all procedure in a specific database
+-----+-----+-----+-----+-----+-----+-----+
| Db | Name | Type | Definer | Modified | Created | Security_type | Comment |
+-----+-----+-----+-----+-----+-----+-----+
| Olympic_Game_2024_22663281 | CAL_TOTAL_MEDALS | PROCEDURE | douserg@localhost | 2024-10-24 17:23:12 | 2024-10-24 17:23:12 | DEFINER | calculation of total medal from a discipline and gender |
| utf8mb4 | utf8mb4_932_ci | utf8mb4_932_ci |
+-----+-----+-----+-----+-----+-----+-----+
| Olympic_Game_2024_22663281 | Counting_Medal_By_discipline | PROCEDURE | douserg@localhost | 2024-10-24 17:24:09 | 2024-10-24 17:24:09 | DEFINER | COUNTING total medal won on a specific discipline |
| utf8mb4 | utf8mb4_932_ci | utf8mb4_932_ci |
+-----+-----+-----+-----+-----+-----+-----+
| Olympic_Game_2024_22663281 | Count_Discipline | PROCEDURE | douserg@localhost | 2024-10-24 17:24:32 | 2024-10-24 17:24:32 | DEFINER |
| utf8mb4 | utf8mb4_932_ci | utf8mb4_932_ci |
+-----+-----+-----+-----+-----+-----+-----+
```

VIEW_TASK1 :

Create VIEW name "MEDALLIST_TEAM " by joining MEDALLIST and TEAM table

where MEDALLIST medal_date medal_type, medalist_id, code_athlete , birth_date, name and

FROM team discipline and events attribute will be included.

```
mysql> CREATE VIEW MEDALLIST_TEAM AS
-> SELECT
->     m.medal_date AS DATE_OF_MEDAL,
->     m.medal_type AS TYPE_OF_MEDAL,
->     m.medalist_id AS MEDALLIST_ID,
->     m.code_athlete AS ATELTE_CODE_M,
->     m.birth_date AS BIRTHDATE,
->     m.name AS NAME,
->     e.code AS TEAM_CODE,
->     e.discipline AS GAMES,
->     e.events AS EVENT_LIST
-> FROM
->     MEDALLIST m
-> JOIN
->     TEAM e
-> ON
->     m.code_team = e.code;
Query OK, 0 rows affected (0.02 sec)
```

By this command we can retrieve data from **view MEDALLIST_TEAM**

```
mysql> SELECT * FROM MEDALLIST_TEAM;
```

MEDAL_ID	DATE_OF_MEDAL	TYPE_OF_MEDAL	MEDALLIST_ID	ATHLETE_CODE_M	BIRTHDATE	NAME	TEAM_CODE	GAMES	EVENT_LIST
2024-07-27	Gold Medal	1	1000100	2000-11-07	China Yang	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	2	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Silver Medal	3	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Silver Medal	4	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Bronze Medal	5	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Bronze Medal	6	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	7	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	8	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	9	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	10	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	11	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	12	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	13	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	14	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	15	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	16	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	17	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	18	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	19	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	20	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	21	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	22	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	23	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating
2024-07-27	Gold Medal	24	1000100	2000-06-11	China Zhao	CHINA	Winter	Figure Skating	Ice Skating

VIEW_TASK2:

update name in **MEDALLIST TABLE** and checked on **VIEW MEDALLIST_TEAM** if the update was successful or not by calling procedure

let's say medallist_id 1159 changed the name and checked

```
mysql> SELECT NAME FROM MEDALLIST_TEAM where MEDALLIST_ID = '1159';
+-----+
| NAME |
+-----+
| van de WIEL Anne |
+-----+
```

1st checking the name in
VIEW MEDALLIST_TEAM
before updating

```
mysql> SELECT name FROM MEDALLIST where medalist_id = '1159';
+-----+
| name |
+-----+
| van de WIEL Anne |
+-----+
1 row in set (0.01 sec)
```

now checking on

MEDALIST TABLE

```
mysql> UPDATE MEDALLIST
-> SET name = 'dhrubo das'
-> WHERE medalist_id = '1159';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

Updating the name IN MEDALLIST TABLE

```
mysql> SELECT NAME FROM MEDALLIST_TEAM where MEDALLIST_ID = '1159';
+-----+
| NAME |
+-----+
| dhrubo das |
+-----+
1 row in set (0.00 sec)

mysql> SELECT name FROM MEDALLIST where medalist_id = '1159';
+-----+
| name |
+-----+
| dhrubo das |
+-----+
1 row in set (0.00 sec)
```

Again checking in MEDALLIST Table & MEDALLIST_TEAM VIEW if the update was successful or not

This query shows all the VIEWS in the DATABASE

```
mysql> SHOW FULL TABLES IN Olympic_Game_2024_22663281 WHERE TABLE_TYPE = 'VIEW';
+-----+-----+
| Tables_in_Olympic_Game_2024_22663281 | Table_type |
+-----+-----+
| MEDALLIST_TEAM | VIEW |
+-----+-----+
1 row in set (0.01 sec)
```

This query shows the Specific VIEW in Details

```
mysql> SHOW CREATE VIEW MEDALLIST_TEAM;
+-----+-----+
| View | Create View |
+-----+-----+
| MEDALLIST_TEAM | CREATE ALGORITHM=UNDEFINED DEFINER=@localhost 'SQL SECURITY DEFINER' VIEW 'MEDALLIST_TEAM' AS select m.medal_date AS 'DATE_OF_MEDAL', m.medal_type AS 'TYPE_OF_MEDAL', m.medalist_id AS 'MEDALLIST_ID', m.code_athlete AS 'ATHLETE_CODE', m.birth_date AS 'BIRTHDATE', m.name AS 'NAME', m.code AS 'TEAM_CODE', m.discipline AS 'GAMES', m.events AS 'EVENT_LIST' from MEDALLIST m join TEAM t on((m.code_team = t.code))) | utf8mb4 | utf8mb4_9324_ci |
+-----+-----+
1 row in set (0.00 sec)
```

Query to drop the VIEW

```
mysql> DROP VIEW MEDALLIST_TEAM;
Query OK, 0 rows affected (0.02 sec)

mysql>
```

7. Connecting Python to MySQL Database

I have this python files to do some operation .

python3 eventdata.py

python3 scheduleData.py

python3 pythonConnect.py

pyhton3 insert.py

7.1 To connect MYSQL Server with python environment need to do this at first like this:


```
STUDENT\22663281@v-2204-hcs-164: /tsclient/dhruv/Downloads/OneDrive_2024-10-23/tr_full_Checked - Copy_PC$ mysql -V
mysql Ver 8.0.39-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
STUDENT\22663281@v-2204-hcs-164: /tsclient/dhruv/Downloads/OneDrive_2024-10-23/tr_full_Checked - Copy_PC$ pip3 install mysql-connector-python
Defaulting to user installation because normal site-packages is not writeable
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.1.0-cp310-cp310-manylinux_2_28_x86_64.whl.metadata (6.6 kB)
  Downloading mysql_connector_python-9.1.0-cp310-cp310-manylinux_2_28_x86_64.whl (34.4 MB)
    34.4/34.4 MB 34.7 MB/s eta 0:00:00
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.1.0
STUDENT\22663281@v-2204-hcs-164: /tsclient/dhruv/Downloads/OneDrive_2024-10-23/tr_full_Checked - Copy_PC$
```

7.2 pythonConnect.py this file shows that we can connect the MYSQL Serve with python environment.

```
1 # this file contains the details of how to connect python to SQL SERVER
2
3 import mysql.connector
4 import getpass
5
6
7 # Read username and password from the user
8 username = input("Enter MySQL username: ")
9 password = getpass.getpass("Enter MySQL password: ")
10
11 try:
12     # Create a connection to the database
13     conn = mysql.connector.connect(
14         host='localhost',
15         user=username,
16         password=password,
17         database='Olympic_Game_2024_22663281'
18     )
19     if conn.is_connected():
20         db_info = conn.get_server_info()
21         print(f"Connected to MySQL Server version {db_info}")
22         cursor = conn.cursor()
23         cursor.execute("SELECT DATABASE();")
24         record = cursor.fetchall()
25         print(f"You're connected to database: {record[0]}")
26
27
28
29 except mysql.connector.Error as err:
30     print(f"Error: {err}")
31 finally:
32     if conn.is_connected():
33         cursor.close()
34         conn.close()
35     print("MySQL connection is closed.")
36
```

```
STUDENT\22663281@v-2204-hcs-164: /tsclient/dhruv/Downloads/OneDrive_2024-10-23/tr_full_Checked - Copy_PC$ python3 pythonConnect.py
Enter MySQL username: dsuser
Enter MySQL password:
Connected to MySQL Server version 8.0.39-0ubuntu0.22.04.1
You're connected to database: ('Olympic_Game_2024_22663281',)
MySQL connection is closed.
```

7.3 evendata.py -- to retrieve all rows of all EVENT table and display using fetchone() command..

```

#eventdata.py -- Add code to retrieve all rows of all EVENT table and display using fetchone() command.
import mysql.connector
import getpass

# Read username and password from the user
username = input("Enter MySQL username: ")
password = getpass.getpass("Enter MySQL password: ")

try:
    # Create a connection to the database
    conn = mysql.connector.connect(
        host='localhost',
        user=username,
        password=password,
        database='Olympic_Game_2024_22663261'
    )
    if conn.is_connected():
        db_info = conn.get_server_info()
        print(f"Connected to MySQL Server version {db_info}")
        cursor = conn.cursor()
        cursor.execute("SELECT DATABASE();")
        record = cursor.fetchone()
        print(f"You're connected to database: {record[0]}")

        # Add the code to retrieve and display rows from the Emp table
        select_query1 = "SELECT * FROM EVENT"
        cursor.execute(select_query1)

        # Get all rows
        rows = cursor.fetchall()

        # Print the first four columns of all rows
        for row in rows:
            print(row[0], row[1], ", ", row[2], ", ", row[3]) # Adjusted to match the first four columns of the EVENT table

except mysql.connector.Error as err:
    print(f"Error: {err}")
finally:
    if conn.is_connected():
        cursor.close()
        conn.close()
    print("MySQL connection is closed.")

```

The above code we imported **library MYSQL connector**. to interacts with SQL server.

The fetchall() statement fetch all the rows from result table . The exception given prints the error if occurred. Then finally close the cursor and then the connection.

```

NAME:STUDENT\22663261@v-2204-acs-164:/tmp/jeetu/Downloads/accs/DevExe_2024-10-22/1st_Fall_Checked - Copy_PC:$ python3 eventdata.py
Enter MySQL username: dsuser
Enter MySQL password:
Connected to MySQL Server version 8.0.39-0ubuntu0.22.04.1
You're connected to database: Olympic_Game_2024_22663261
Men's Individual Archery , ARC , 1
Women's Individual Archery , ARC , 2
Men's Team Archery , ARC , 3
Women's Team Archery , ARC , 4
Mixed Team Archery , ARC , 5
Men's Team Artistic Gymnastics , GAR , 6
Men's All-Around Artistic Gymnastics , GAR , 7
Men's Floor Exercise Artistic Gymnastics , GAR , 8
Men's Pommel Horse Artistic Gymnastics , GAR , 9
Men's Rings Artistic Gymnastics , GAR , 10
Men's Vault Artistic Gymnastics , GAR , 11
Men's Parallel Bars Artistic Gymnastics , GAR , 12
Men's Horizontal Bar Artistic Gymnastics , GAR , 13
Women's Team Artistic Gymnastics , GAR , 14
Women's All-Around Artistic Gymnastics , GAR , 15
Women's Vault Artistic Gymnastics , GAR , 16
Women's Uneven Bars Artistic Gymnastics , GAR , 17
Women's Balance Beam Artistic Gymnastics , GAR , 18
Women's Floor Exercise Artistic Gymnastics , GAR , 19
Duet Artistic Swimming , SMA , 20
Team Artistic Swimming , SMA , 21
Men's 100m Athletics , ATH , 22
Men's 200m Athletics , ATH , 23
Men's 400m Athletics , ATH , 24
Men's 800m Athletics , ATH , 25
Men's 1500m Athletics , ATH , 26
Men's 5000m Athletics , ATH , 27
Men's 10,000m Athletics , ATH , 28
Men's Marathon Athletics , ATH , 29
Men's 3000m Steeplechase Athletics , ATH , 30
Men's 110m Hurdles Athletics , ATH , 31
Men's 400m Hurdles Athletics , ATH , 32
Men's High Jump Athletics , ATH , 33
Men's Pole Vault Athletics , ATH , 34
Men's Long Jump Athletics , ATH , 35
Men's Triple Jump Athletics , ATH , 36
Men's Shot Put Athletics , ATH , 37

```

7.4 schdeule.py this pyhton file retrieve the data of SCEHEDULE table such as end_date, start_date, vanue_code using where conditon status = CANCELLED


```

# schedule.py this python file retrieve the data of SCHEDULE table such as end_date, start_date, venue_code using where condition status = CANCELLED
import mysql.connector
import getpass

# Read username and password from the user
username = input("Enter MySQL username: ")
password = getpass.getpass("Enter MySQL password: ")

try:
    # Create a connection to the database
    conn = mysql.connector.connect(
        host='localhost',
        user=username,
        password=password,
        database='Olympic_Game_2024_22663281'
    )
    if conn.is_connected():
        db_info = conn.get_server_info()
        print(f"Connected to MySQL Server version {db_info}")
        cursor = conn.cursor()
        cursor.execute("SELECT DATABASE()")
        record = cursor.fetchall()
        print(f"You're connected to database: {record[0]}")

        # Retrieve and display the employee's details using the hardcoded employee number
        select_query = """SELECT start_date, end_date, venue_code
                        FROM SCHEDULE WHERE
                        STATUS = 'CANCELLED' """

        cursor.execute(select_query)

        # Fetch and display the result
        rows = cursor.fetchall()
        if rows:
            for row in rows:
                print(f"start date: {row[0]}, end date: {row[1]}, venue code: {row[2]}")
        else:
            print("No schedule events with status found.")
except mysql.connector.Error as err:
    print(f"Error: {err}")
finally:
    if conn.is_connected():
        cursor.close()
        conn.close()
    print("MySQL connection is closed.")

```

Shows me the result of start_date, end date and venue code that have been cancelled

```

STUDENT\22663281@v-2204-bcs-164:~/tsclient/shrub/Downloads/OneDrive_2024-10-23/tr_full_checked - Copy_PC$ python3 scheduleData.py
Enter MySQL username: dsuser
Enter MySQL password:
Connected to MySQL Server version 8.0.39-Bubuntu0.22.04.1
You're connected to database: ('Olympic Game 2024 22663281',)
start date: 2024-07-29 14:30:00, end date: 2024-07-29 14:30:00, venue code: CPL
start date: 2024-07-29 20:00:00, end date: 2024-07-29 20:00:00, venue code: CPL
start date: 2024-07-30 20:50:00, end date: 2024-07-30 20:50:00, venue code: CPL
start date: 2024-07-31 15:20:00, end date: 2024-07-31 15:20:00, venue code: CPL
start date: 2024-08-01 19:36:00, end date: 2024-08-01 19:57:00, venue code: MAM
start date: 2024-08-01 19:56:00, end date: 2024-08-01 20:17:00, venue code: MAM
start date: 2024-08-01 20:07:00, end date: 2024-08-01 20:28:00, venue code: MAM
start date: 2024-08-01 20:27:00, end date: 2024-08-01 20:48:00, venue code: MAM
start date: 2024-08-01 23:38:00, end date: 2024-08-01 23:59:00, venue code: MAM
start date: 2024-08-01 23:38:00, end date: 2024-08-01 23:59:00, venue code: MAM
start date: 2024-08-02 00:09:00, end date: 2024-08-02 00:30:00, venue code: MAM
start date: 2024-08-02 00:28:00, end date: 2024-08-02 00:49:00, venue code: MAM
start date: 2024-08-02 00:28:00, end date: 2024-08-02 00:49:00, venue code: MAM
start date: 2024-08-02 00:40:00, end date: 2024-08-02 01:01:00, venue code: MAM
start date: 2024-08-02 00:59:00, end date: 2024-08-02 01:20:00, venue code: MAM
start date: 2024-08-02 01:30:00, end date: 2024-08-02 01:51:00, venue code: MAM
start date: 2024-08-02 02:01:00, end date: 2024-08-02 02:22:00, venue code: MAM
start date: 2024-08-05 19:23:00, end date: 2024-08-05 20:14:00, venue code: MAM
start date: 2024-08-05 20:45:00, end date: 2024-08-05 21:36:00, venue code: MAM
start date: 2024-08-05 21:50:00, end date: 2024-08-05 22:41:00, venue code: MAM
start date: 2024-08-06 20:29:00, end date: 2024-08-06 21:19:00, venue code: MAM
start date: 2024-08-06 21:36:00, end date: 2024-08-06 22:26:00, venue code: MAM
start date: 2024-08-07 18:03:00, end date: 2024-08-07 18:20:00, venue code: MAM
start date: 2024-08-07 18:23:00, end date: 2024-08-07 18:40:00, venue code: MAM
start date: 2024-08-07 18:43:00, end date: 2024-08-07 19:00:00, venue code: MAM
start date: 2024-08-07 18:48:00, end date: 2024-08-07 19:05:00, venue code: MAM
start date: 2024-08-07 19:13:00, end date: 2024-08-07 19:30:00, venue code: MAM
start date: 2024-08-07 19:23:00, end date: 2024-08-07 19:40:00, venue code: MAM
start date: 2024-08-07 19:38:00, end date: 2024-08-07 19:55:00, venue code: MAM
start date: 2024-08-07 20:03:00, end date: 2024-08-07 20:20:00, venue code: MAM
start date: 2024-08-07 20:03:00, end date: 2024-08-07 20:20:00, venue code: MAM
start date: 2024-08-07 20:43:00, end date: 2024-08-07 21:00:00, venue code: MAM
start date: 2024-08-07 22:00:00, end date: 2024-08-07 22:17:00, venue code: MAM
start date: 2024-08-07 22:00:00, end date: 2024-08-07 22:17:00, venue code: MAM
start date: 2024-08-07 22:00:00, end date: 2024-08-07 22:17:00, venue code: MAM

```

7.4 : insert.py – is the file where we are inserting new value into EVENT tble. The user can insert a new event, sports sports_code according to their choice and then can cheked it into the **MYSQL SERVER** like this.

Here user can ask the event , sport_name sport code and the file will excetued it .

```
# insert.py ---- is the file that do insertion in python into the EVENT table
import mysql.connector
import getpass

# asking user for useranme and password to be connected
username = input("Enter your MySQL username: ")
password = getpass.getpass("Enter MySQL password: ")

# here, trying to connected with MYSQL server
try:
    conn = mysql.connector.connect(
        host='localhost',
        user=username,
        password=password,
        database='Olympic_Game_2024_22663281'
    )

    # here, creating a cursor object suin cursor () method
    cursor = conn.cursor()

    # here, inserting the new data inTO EVENT TABLE OF MYSQL SERVER
    insert_query = """
    INSERT INTO EVENT (event, sport, sport_code)
    VALUES (%s, %s, %s)
    """

    # We can ask user to give event_name, sport_name, sport_code according to their choice
    event_name = input("Hi!!... plz Enter the event name: ")
    sport_name = input("Hi!!...plz Enter the sport name: ")
    sport_code = input("HI there!! plz...Enter the sport code: ")

    # to store the value from user input
    NewRow = (event_name, sport_name, sport_code)

    # command to exeute SQL
    cursor.execute(insert_query, NewRow)

    # by typing this commit the changes in the DATABASES
    conn.commit()
    print("WOW!!!!!! Row inserted successfully.")

except mysql.conn.Error as er:
    print(f"Failed to insert record into MySQL table: {er}")

    # rollback of any error
    if conn:
        conn.rollback()

finally:
    # Close the cursor and connection
    if cursor:
        cursor.close()
    if conn:
        conn.close()
```

```
STUDENT\22663281@v-2204-hcs-121:~/tsclient/22663281/OneDrive - Curtin/tr_full_Checkd - Copy_PCC$ python3 s.py
Enter your MySQL username: dsuser
Enter your MySQL password: userCreateSQL
Enter the event name: Women's 81kg
Enter the sport name: WEIGHTLIFTING
Enter the sport code: WLF
Row inserted successfully.
```

Sorry, I renamed the file s.py as insert.py which contains the same information as s.py.

Now we are checking if the data is inserted or not . by checking into MYSQL SERVER.

Step:1.

```
STUDENT\22663281@v-2204-hcs-121:~/tsclient/22663281/OneDrive - Curtin/tr_full_Checkd - Copy_PCC$ mysql -V
mysql Ver 8.0.39-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
```

Step2:

```
STUDENT\22663281@v-2204-hcs-121:~/tsclient/22663281/OneDrive - Curtin/tr_full_Checkd - Copy_PCC$ mysql --local-infile=1 -u dsuser -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 24
Server version: 8.0.39-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Step3 :SELECT the database

```
mysql> use Olympic_Game_2024_22663281;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

Step4 : retrieving all data from EVENT table


```
mysql> SELECT * FROM EVENT;
```

event	sport	sport_code
Men's Individual	Archery	ARC
Women's Individual	Archery	ARC
Men's Team	Archery	ARC
Women's Team	Archery	ARC
Mixed Team	Archery	ARC
Men's Team	Artistic Gymnastics	GAR
Men's All-Around	Artistic Gymnastics	GAR
Men's Floor Exercise	Artistic Gymnastics	GAR
Men's Pommel Horse	Artistic Gymnastics	GAR
Men's Rings	Artistic Gymnastics	GAR
Men's Vault	Artistic Gymnastics	GAR
Men's Parallel Bars	Artistic Gymnastics	GAR
Men's Horizontal Bar	Artistic Gymnastics	GAR
Women's Team	Artistic Gymnastics	GAR
Women's All-Around	Artistic Gymnastics	GAR
Women's Vault	Artistic Gymnastics	GAR
Women's Uneven Bars	Artistic Gymnastics	GAR
Women's Balance Beam	Artistic Gymnastics	GAR
Women's Floor Exercise	Artistic Gymnastics	GAR
Duet	Artistic Swimming	SWA
Team	Artistic Swimming	SWA
Men's 100m	Athletics	ATH
Men's 200m	Athletics	ATH
Men's 400m	Athletics	ATH
Men's 800m	Athletics	ATH

Men	Volleyball	VVO
Women	Volleyball	VVO
Men	Water Polo	WPO
Women	Water Polo	WPO
Women's 49kg	Weightlifting	WLF
Women's 59kg	Weightlifting	WLF
Women's 71kg	Weightlifting	WLF
Women's 81kg	Weightlifting	WLF
Women's +81kg	Weightlifting	WLF
Men's 61kg	Weightlifting	WLF
Men's 73kg	Weightlifting	WLF
Men's 89kg	Weightlifting	WLF
Men's 102kg	Weightlifting	WLF
Men's +102kg	Weightlifting	WLF
Men's Greco-Roman 60kg	Wrestling	WRE
Men's Greco-Roman 67kg	Wrestling	WRE
Men's Greco-Roman 77kg	Wrestling	WRE
Men's Greco-Roman 87kg	Wrestling	WRE
Men's Greco-Roman 97kg	Wrestling	WRE
Men's Greco-Roman 130kg	Wrestling	WRE
Women's Freestyle 50kg	Wrestling	WRE
Women's Freestyle 53kg	Wrestling	WRE
Women's Freestyle 57kg	Wrestling	WRE
Women's Freestyle 62kg	Wrestling	WRE
Women's Freestyle 68kg	Wrestling	WRE
Women's Freestyle 76kg	Wrestling	WRE
Men's Freestyle 57kg	Wrestling	WRE
Men's Freestyle 65kg	Wrestling	WRE
Men's Freestyle 74kg	Wrestling	WRE
Men's Freestyle 86kg	Wrestling	WRE
Men's Freestyle 97kg	Wrestling	WRE
Men's Freestyle 125kg	Wrestling	WRF
Women's 81kg	WEIGHTLIFTING	WLF

here the **BLUE marked** row is the newly inserted data that we did by using python insert.py. So it can be said that the insert.py file is working correctly. It is successfully inserting the data using python environment into MySQL SERVER.

8. Discussion

The implementation and the execution of Olympic game 2024 was really interesting as well was challenging. The whole part was sorting data and implementing the queries according to individual condition and create the queries. It was like making your own queries and then get the output based on that using the MYSQL theories of joining, advance features like PROCEDURE and VIEWS.

While implementing and sorting my data to get my output I also got knowledge of so many countries coach name, player name, and which country got the most medal in specific game and a lot more information. that I was unaware of.

Moreover, it was also challenging, because sorting data was not easy and to create the queries was also a long process as there are many more data and many information that I can also work. I might be work well, but happy t to present my data of Olympic games 2024.Overall, I was happy and blessed to work with this which enrich my knowledge of SQL