Database project - deliverable 2

Arthur Giroux 205443

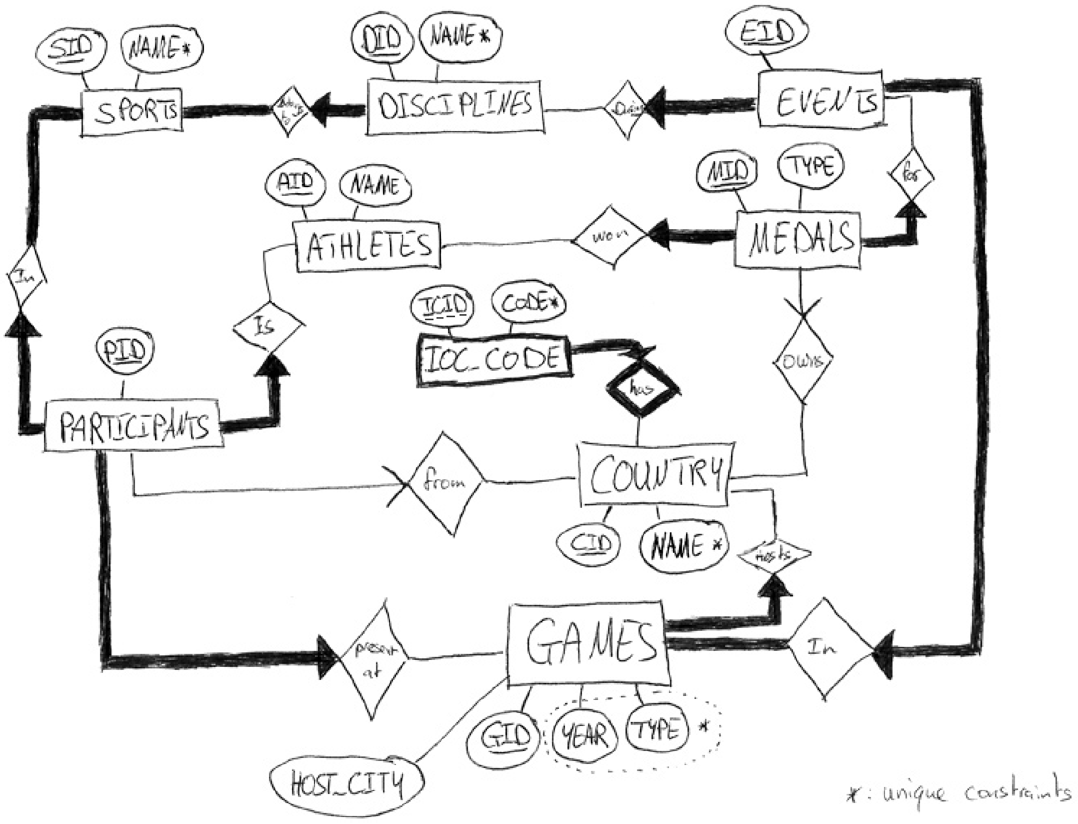
Colla Rensch 205814

Valentin Matter 203447

April 21st 2013

Deliverable 1

1 ER model



2 Tables creation

```
CREATE TABLE COUNTRIES
(
  CID INTEGER NOT NULL
, NAME VARCHAR2(50) NOT NULL
, PRIMARY KEY ( CID )
, CONSTRAINT unique_country_name UNIQUE ( NAME )
);
```

```sql

CREATE TABLE IOC_CODE
(
  ICID INTEGER NOT NULL
, CODE CHAR(3) NOT NULL
, CID INTEGER NOT NULL
, PRIMARY KEY ( ICID )
, FOREIGN KEY ( CID ) REFERENCES COUNTRIES( CID ) ON DELETE CASCADE
, CONSTRAINT unique_ioc_code_name UNIQUE ( CODE )
);

CREATE TABLE SPORTS
(
  SID INTEGER NOT NULL
, NAME VARCHAR2(100) NOT NULL
, PRIMARY KEY ( SID )
, CONSTRAINT unique_sport_name UNIQUE ( NAME )
);

CREATE TABLE ATHLETES
(
  AID INTEGER NOT NULL
, NAME VARCHAR2(200) NOT NULL
, PRIMARY KEY ( AID )
);

CREATE TABLE DISCIPLINES
(
  DID INTEGER NOT NULL
, NAME VARCHAR2(200) NOT NULL
, SID INTEGER NOT NULL
, PRIMARY KEY ( DID )
, FOREIGN KEY ( SID ) REFERENCES SPORTS( SID )
, CONSTRAINT unique_discipline UNIQUE ( NAME, SID )
);

CREATE TABLE GAMES
(
  GID INTEGER NOT NULL
, YEAR INTEGER NOT NULL
, TYPE VARCHAR2(50) NOT NULL
, HOST_CITY VARCHAR2(200) NOT NULL
, CID INTEGER NOT NULL
, PRIMARY KEY ( GID )
, FOREIGN KEY ( CID ) REFERENCES COUNTRIES( CID )
, CONSTRAINT unique_game UNIQUE ( YEAR, TYPE )
);

CREATE TABLE EVENTS
(
  EID INTEGER NOT NULL
, GID INTEGER NOT NULL
, DID INTEGER NOT NULL
, PRIMARY KEY ( EID )
, FOREIGN KEY ( GID ) REFERENCES GAMES( GID )
, FOREIGN KEY ( DID ) REFERENCES DISCIPLINES( DID )
, CONSTRAINT unique_event UNIQUE ( GID, DID )
);

CREATE TABLE PARTICIPANTS
(
  PID INTEGER NOT NULL
, AID INTEGER NOT NULL
, CID INTEGER
, GID INTEGER NOT NULL
, SID INTEGER NOT NULL
, PRIMARY KEY ( PID )
, FOREIGN KEY ( AID ) REFERENCES ATHLETES(AID)
, FOREIGN KEY ( CID ) REFERENCES COUNTRIES(CID)
, FOREIGN KEY ( GID ) REFERENCES GAMES(GID)
, FOREIGN KEY ( SID ) REFERENCES SPORTS(SID)
, CONSTRAINT unique_participant UNIQUE ( AID, CID, GID, SID )
);

CREATE TABLE MEDALS
(
  MID INTEGER NOT NULL
, TYPE VARCHAR2(50) NOT NULL
, CID INTEGER
, EID INTEGER NOT NULL
, AID INTEGER NOT NULL
, PRIMARY KEY ( MID )
, FOREIGN KEY ( CID ) REFERENCES COUNTRIES(CID)
, FOREIGN KEY ( AID ) REFERENCES ATHLETES(AID)
, FOREIGN KEY ( EID ) REFERENCES EVENTS(EID)
, CONSTRAINT unique_medalist UNIQUE ( CID, EID, AID )
);
```

---

3 Remarks

– Athletes is the entity that stores the information of an athlete who can then participate in multiple sports or games. Which means that if an Athlete competes twice he will have only one entry in the ATHLETES table but two in the PARTICIPANTS table.
– Each game should at least have one event otherwise nothing happened during he games. The same applies to sports, a sport must have at least one participant, otherwise the sport never took place during any games.
– Countries, sports and disciplines must have a unique name as the opposite would have no sense. For games it is the pair (YEAR,TYPE) that must be unique.
– Due to problem with their federations some athletes may present themselves without representing a country.
– Some medals could not be associated to a country as some athletes aren't as for the above point.
– The numbers of countries, athletes and events for a given game isn't stored in the GAMES entity as they can be easily computed using some COUNT query.
– The names of the events, disciplines and games are not stored as they are only the concatenation of information stored in other tables.

Deliverable 2

Modifications

We had to slightly modify our model as we forced each athlete to have a binding participant, but we noticed that there is a quite big amount of them that do not. Therefore, we decided to slightly change our ER-model as to allow athletes to not have a binding participant. No change to the table creation code was needed as it is not possible to represent the "at least" constraint.

Data import

We chose to import the data using Java. We decided that any data that would generate a non-existent foreign key would be dropped was would any inconsistent incomplete data (i.e. medals without color)

Queries

A) Simple query using multiple ANDs

```sql
SELECT DISTINCT A.NAME
FROM ATHLETES A, MEDALS M1, MEDALS M2, EVENTS E1, EVENTS E2, GAMES G1, GAMES
    G2
WHERE A.AID = M1.AID
AND A.AID = M2.AID
AND M1.MID <> M2.MID
AND E1.EID = M1.EID
AND E2.EID = M2.EID
AND G1.GID = E1.GID
AND G2.GID = E2.GID
AND G1.TYPE <> G2.TYPE
```

B) We have an outer-query which seeks gold medalist in sports that appear in the nested query, which computes all sports that have appeared only once

```sql
SELECT A.NAME AS ANAME, S.NAME AS SNAME
FROM SPORTS S, DISCIPLINES D, EVENTS E, ATHLETES A, MEDALS M
WHERE A.AID = M.AID
AND M.TYPE = 'Gold'
AND M.EID = E.EID
AND E.DID = D.DID
AND D.SID = S.SID
AND S.SID IN (
  SELECT S2.SID
  FROM SPORTS S2, DISCIPLINES D2, EVENTS E2, GAMES G
  WHERE S2.SID = D2.SID
  AND D2.DID = E2.DID
  AND E2.GID = G.GID
  GROUP BY S2.SID
  HAVING Count(*)=1
)
ORDER BY A.NAME
```

C) We retrieve the minimum year in which each country won its first medal using a subquery and then use a simple query to get the place hosting the corresponding games.

```sql
SELECT DISTINCT C.NAME, G.HOST_CITY
FROM COUNTRIES C, MEDALS M, EVENTS E, GAMES G, (
  SELECT C2.CID, MIN( G2.YEAR ) AS MIN_YEAR
  FROM COUNTRIES C2, MEDALS M2, EVENTS E2, GAMES G2
  WHERE C2.CID = M2.CID
  AND M2.EID = E2.EID
  AND E2.GID = G2.GID
  GROUP BY C2.CID
) TMP
WHERE C.CID = M.CID
AND M.EID = E.EID
AND E.GID = G.GID
AND C.CID = TMP.CID
AND G.YEAR = TMP.MIN_YEAR
ORDER BY C.NAME
```

D) We unite (UNION) two same queries using "Summer" for one and "Winter" for the other and compute the number of medals for each country given the type (Summer or Winter), we then order them despondingly and limit the table to 1

```
(
  SELECT COUNT(M.CID) AS MAXIMUM, C.NAME, G.TYPE
  FROM COUNTRIES C, GAMES G, EVENTS E, MEDALS M
  WHERE C.CID = M.CID
  AND M.EID = E.EID
  AND E.GID = G.GID
  AND G.TYPE = 'Summer'
  GROUP BY C.NAME
  ORDER BY MAXIMUM DESC
  LIMIT 1
) UNION (
  SELECT COUNT(M.CID) AS MAXIMUM, C.NAME, G.TYPE
  FROM COUNTRIES C, GAMES G, EVENTS E, MEDALS M
  WHERE C.CID = M.CID
  AND M.EID = E.EID
  AND E.GID = G.GID
  AND G.TYPE = 'Winter'
  GROUP BY C.NAME
  ORDER BY MAXIMUM DESC
  LIMIT 1
)
```

E) Simple GROUP BY + HAVING query

```
SELECT G.HOST_CITY
FROM GAMES G
GROUP BY G.HOST_CITY
HAVING COUNT(*) > 1
ORDER BY G.HOST_CITY
```

F) We use two table of participants and two tables for countries and then just use ANDs to make find the athletes that competed for at least two countries

```
SELECT DISTINCT(A.NAME)
FROM ATHLETES A, PARTICIPANTS P1, PARTICIPANTS P2, COUNTRIES C1, COUNTRIES C2
WHERE A.AID = P1.AID
AND A.AID = P2.AID
AND P1.PID <> P2.PID
AND P1.CID = C1.CID
AND P2.CID = C2.CID
AND C1.CID <> C2.CID
ORDER BY A.NAME
```

G) The subquery computes the participants count for each country for a particular games. Then, for each game, the outer-query finds the countries having a participants count greater than all the results of the subquery.

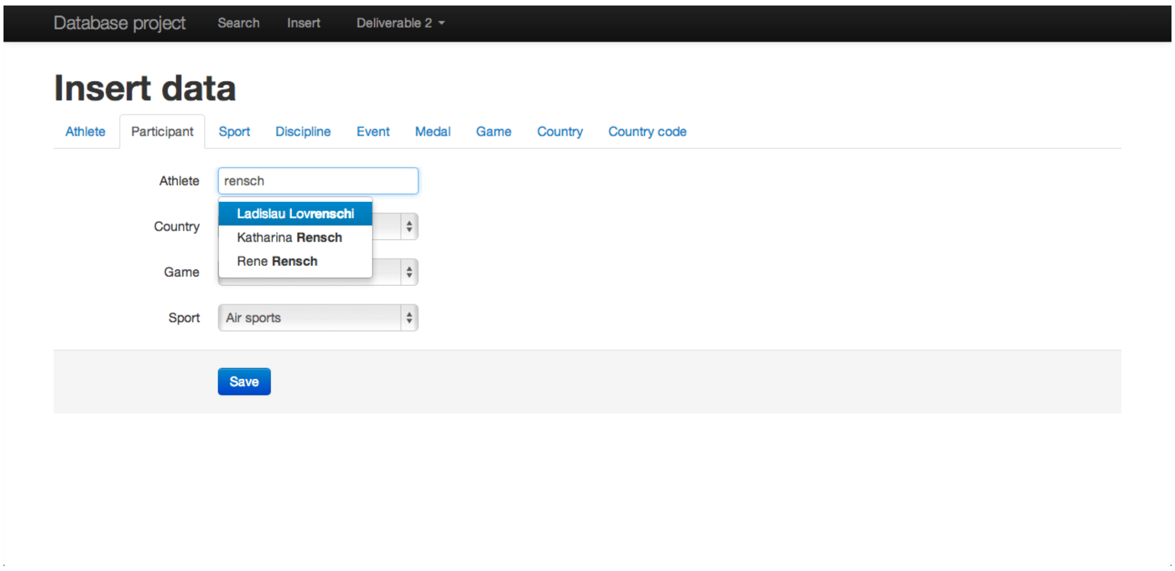
```
SELECT G.YEAR, G.TYPE, C.NAME, COUNT(*) AS COUNT
FROM COUNTRIES C, PARTICIPANTS P, GAMES G
WHERE G.GID = P.GID AND C.CID=P.CID
GROUP BY P.GID, P.CID
HAVING COUNT(*) >= ALL (
  SELECT COUNT(*)
  FROM PARTICIPANTS P2
  WHERE P.GID=P2.GID
  GROUP BY P2.GID, P2.CID
)
ORDER BY G.YEAR
```

H) We simply take the COUNTRIES and use a nested query to delete all entries that do not appear in the MEDALS table

```
SELECT DISTINCT C.NAME
FROM COUNTRIES C
WHERE C.CID NOT IN (
    SELECT M.CID
    FROM MEDALS M
    WHERE M.CID IS NOT NULL
)
```

Front-end

Our web front-end is available at `http://db.tamere.ch/`. It was made using PHP and MySQL.

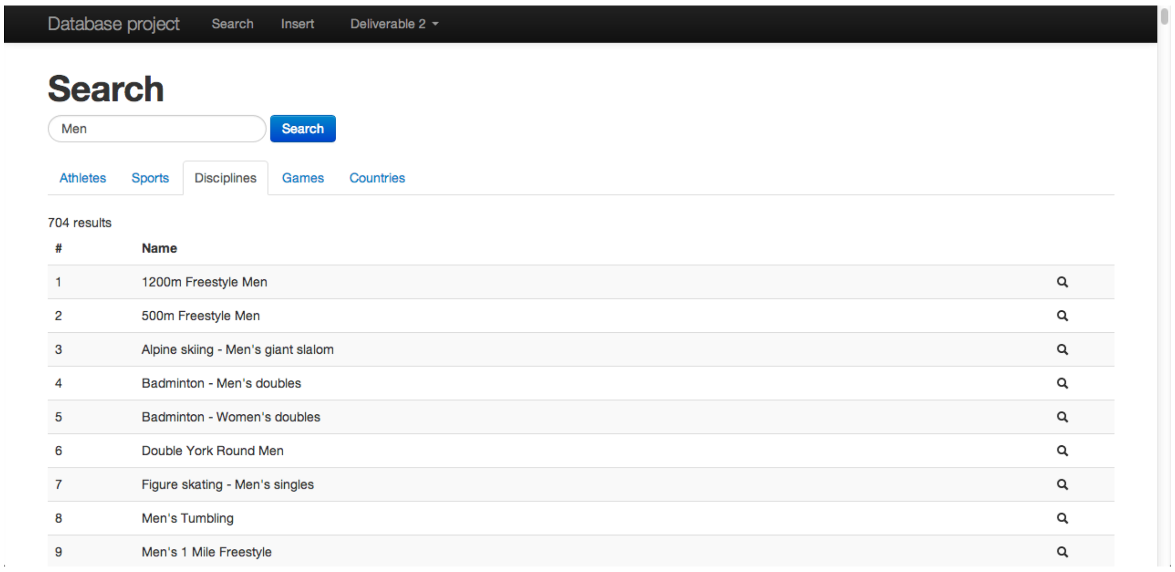


Insert data

Athlete | Participant | Sport | Discipline | Event | Medal | Game | Country | Country code

Athlete	rensch
Country	Ladislau Lovrenschi Katharina **Rensch** Rene **Rensch**
Game	
Sport	Air sports

The insertion page



Search

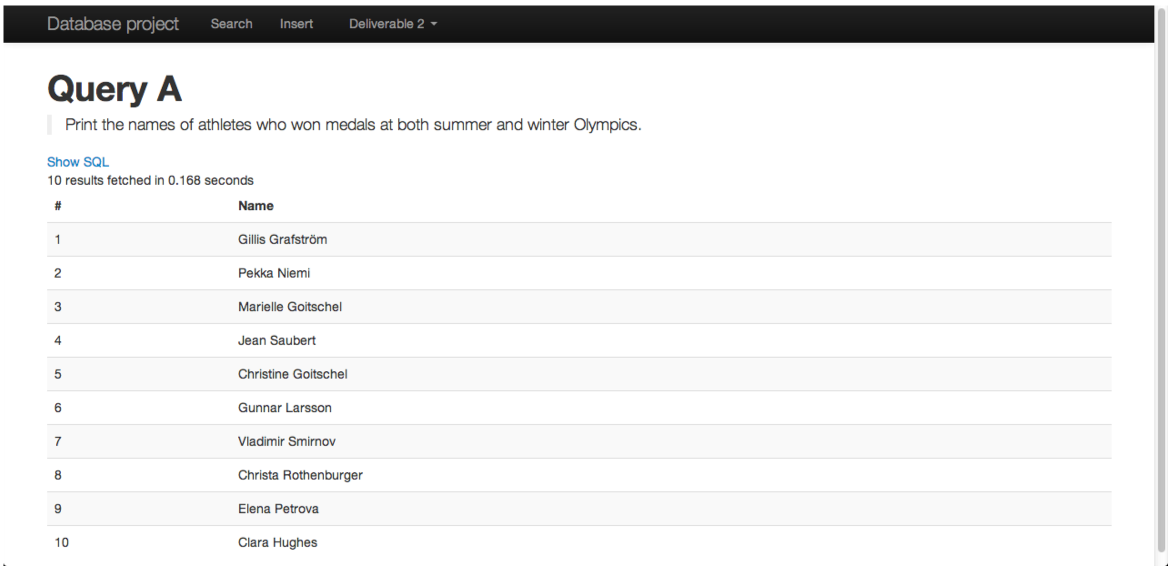
Men

Athletes | Sports | Disciplines | Games | Countries

704 results

#	Name
1	1200m Freestyle Men
2	500m Freestyle Men
3	Alpine skiing - Men's giant slalom
4	Badminton - Men's doubles
5	Badminton - Women's doubles
6	Double York Round Men
7	Figure skating - Men's singles
8	Men's Tumbling
9	Men's 1 Mile Freestyle

The search page



Query A

Print the names of athletes who won medals at both summer and winter Olympics.

Show SQL

10 results fetched in 0.168 seconds

#	Name
1	Gillis Grafström
2	Pekka Niemi
3	Marielle Goitschel
4	Jean Saubert
5	Christine Goitschel
6	Gunnar Larsson
7	Vladimir Smirnov
8	Christa Rothenburger
9	Elena Petrova
10	Clara Hughes

The result of one of the queries

Deliverable 3

```
CREATE VIEW MEDALS_UNIQUE AS (
  SELECT MIN(M.MID) AS MID, M.CID, M.TYPE, M.EID
  FROM MEDALS M
  WHERE M.TID IS NOT NULL
  GROUP BY M.TID
) UNION (
  SELECT M.MID, M.CID, M.TYPE, M.EID
  FROM MEDALS M
  WHERE M.TID IS NULL
)
```