



SQL GROUP PROJECT

Beijing 2022 Olympic Winter Games

INFO8075-22W-Sec1-SQL and Data Analysis

Conestoga College, Doon, Kitchener

Team Members	Student Id
Amritjot Kaur	8798254
Palash Dakare	8716688
Ali Mustafa	8797917
Sajid Poswal	8760931

Table of Contents

Introduction:	3
Data-sets:	3
Relational Schema:	4
Objectives and Goals:	5
Analysis:	6
References:	17

Introduction

The 2022 Winter Olympics, officially known as the XXIV Olympic Winter Games, were an international winter multi-sport event held in Beijing, China, and adjacent areas from February 4 to February 20, 2022, with competition in selected events commencing on February 2, 2022.

Following the 2018 Winter Olympics in Pyeongchang, South Korea, and the 2020 Summer Olympics in Tokyo, Japan, Beijing was chosen as the host city in 2015 at the 128th IOC Session in Kuala Lumpur, Malaysia. This will be Beijing's second time hosting the Olympics, and the last of three consecutive Olympics hosted in East Asia following the 2018 Winter Olympics in Pyeongchang, South Korea, and the 2020 Summer Olympics in Tokyo, Japan. Beijing became the first city to host both the Summer and Winter Olympics after hosting the Summer Olympics in 2008. The Games were held mostly in Beijing, its suburb Yanqing District, and Zhangjiakou, with some events (such as the ceremonies and curling) repurposing venues built for Beijing 2008. (Such as Beijing National Stadium and the Beijing National Aquatics Centre).

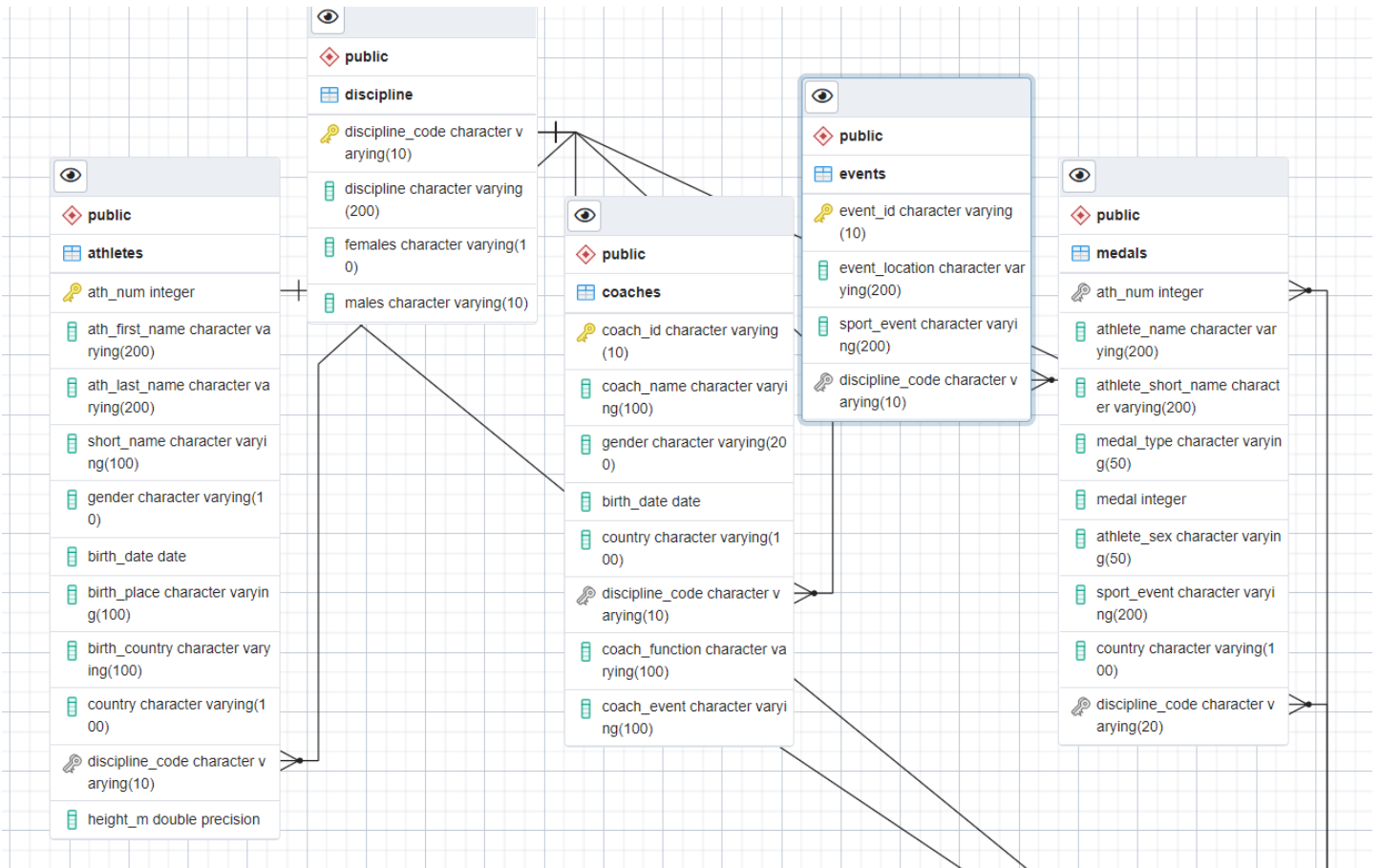
The Games included a total of 109 events across 15 sports, including Olympic debuts for big air freestyle skiing and mono bob, as well as numerous new mixed categories. The Games included 2,871 participants from 91 countries, with Haiti and Saudi Arabia making their Winter Olympic debuts.

Data-sets

- ✚ This is the Olympic Winter Games dataset that describes medals, results, athletes, coaches, and other records for Beijing 2020.
- ✚ Almost 3000 Athletes, 200 Coaches and Technical Officials (with some personal data: date and place of birth, height, etc.), 600 Medals, 600 Events, 15 Disciplines, and Results (Hockey & Curling) of the XXIV Olympic Winter Games you can find in the data.
- ✚ This project includes 5 different tables for analysis as follows:

Athletes	Personal information about all athletes
Coaches	Personal information about all coaches
Entries discipline	Athletes' entries (grouped by discipline)
Events	All events that had a place (qualifications are included)
Medals	General information on all athletes who won a medal

Relational Schema



Objectives and Goals

The sole purpose of Data Analysis is to get meaningful information, and a process of combining multiple entities can be performed by SQL aggregation query. Aggregation is a deterministic function, and it involves calculation for a set of values giving a single entity.

Some of the Goals will be as follows:

1- Importing and Cleaning the Datasets:

- a) Creating tables using external files.
- b) Identifying duplicates from the “Athletes” dataset.
- c) Removing duplicates from the “Athletes” dataset.

2- Display the athletes’ names and their country’s names for all gold medal winners.

3- Calculate the total athletes based on different disciplines.

4- Create a new column that assigns a value based on the Heights of athletes:

- If the number of medals is greater than 1.8 meters, assign Tall.
- If the number of medals is between 1.6 to 1.79 meters, assign Medium Height.
- If the number of medals is below 1.59, assign Short.

5- How many Golds, Silver, and Bronze medals are there in total?

6- Display the coach details of gold medal winners.

7- Display the information of the location of the event where the athletes won a maximum of 3 or more gold medals.

8- Create a view to display details of athletes participating in the doubles event and winning the gold medal.

9- Display country-wise total medals for all the sports events, as well as the grand total.

10- Display the youngest players as per their disciplines. Acquire the data only for those, whose birth date is given in the data.

11- Show athletes’ full names from the athletes’ table and the total number of medals they won.

12- Event-Wise Gold, Silver & Bronze Medals.

13- Create a materialized view for all the participants who played in singles events.

14- Write a nested query for information on all the female athletes who won medals for Canada in ice hockey.

Analysis

1- Creating tables using external files.

Table 1- CREATE TABLE DISCIPLINE (DISCIPLINE_CODE VARCHAR(10) PRIMARY KEY, DISCIPLINE VARCHAR(200), FEMALES VARCHAR (10), MALES VARCHAR (10));

COPY DISCIPLINE (DISCIPLINE, DISCIPLINE_CODE, FEMALES, MALES)

FROM 'D:/SQL PROJECT/OLYMPICS WINTER GAMES/ENTRIES_DISCIPLINE.CSV'

DELIMITER ','

CSV HEADER;

SELECT * FROM DISCIPLINE;

Table 2- CREATE TABLE ATHLETES(ATH_NUM INTEGER PRIMARY KEY, ATH_NAME VARCHAR(200), SHORT_NAME VARCHAR (100), GENDER VARCHAR(10), BIRTH_DATE DATE, BIRTH_PLACE VARCHAR (100), BIRTH_COUNTRY VARCHAR(100), COUNTRY VARCHAR(100), DISCIPLINE_CODE VARCHAR(10)NOT NULL, HEIGHT_M FLOAT, FOREIGN KEY(DISCIPLINE_CODE) REFERENCES DISCIPLINE (DISCIPLINE_CODE) ON UPDATE CASCADE ON DELETE CASCADE);

COPY ATHLETES (ATH_NUM, ATH_NAME, SHORT_NAME, GENDER, BIRTH_DATE, BIRTH_PLACE, BIRTH_COUNTRY, COUNTRY, DISCIPLINE_CODE, HEIGHT_M)

FROM 'D:/SQL PROJECT/OLYMPICS WINTER GAMES/ATHLETES.CSV'

DELIMITER ','

CSV HEADER;

SELECT * FROM ATHLETES;

Table 3- CREATE TABLE COACHES(COACH_ID VARCHAR(10) PRIMARY KEY, COACH_NAME VARCHAR(100), GENDER VARCHAR(200), BIRTH_DATE DATE, COUNTRY VARCHAR(100), DISCIPLINE_CODE VARCHAR(10)NOT NULL, COACH_FUNCTION VARCHAR(100), COACH_EVENT VARCHAR(100), FOREIGN KEY(DISCIPLINE_CODE) REFERENCES DISCIPLINE (DISCIPLINE_CODE) ON UPDATE CASCADE ON DELETE CASCADE);

COPY COACHES (COACH_ID, COACH_NAME, GENDER, BIRTH_DATE, COUNTRY, DISCIPLINE_CODE, COACH_FUNCTION, COACH_EVENT)

FROM 'D:/SQL PROJECT/OLYMPICS WINTER GAMES/COACHES.CSV'

DELIMITER ','

CSV HEADER;

SELECT * FROM COACHES;

Table 4- CREATE TABLE EVENTS (EVENT_ID VARCHAR(10) PRIMARY KEY, EVENT_LOCATION VARCHAR(200) , SPORT_EVENT VARCHAR(200), DISCIPLINE_CODE VARCHAR(10)NOT NULL, FOREIGN KEY(DISCIPLINE_CODE) REFERENCES DISCIPLINE (DISCIPLINE_CODE) ON UPDATE CASCADE ON DELETE CASCADE);

COPY EVENTS (EVENT_ID, EVENT_LOCATION, SPORT_EVENT, DISCIPLINE_CODE)

FROM 'D:/SQL PROJECT/OLYMPICS WINTER GAMES/EVENTS.CSV'

DELIMITER ','

CSV HEADER;

SELECT * FROM EVENTS;

Table 5- CREATE TABLE MEDALS (ATH_NUM INTEGER NOT NULL, ATHLETE_NAME VARCHAR(200) ,
ATHLETE_SHORT_NAME VARCHAR(200), MEDAL_TYPE VARCHAR(50), MEDAL INTEGER, ATHLETE_SEX VARCHAR(50),
SPORT_EVENT VARCHAR (200), COUNTRY VARCHAR(100), DISCIPLINE_CODE VARCHAR (20) NOT NULL,
FOREIGN KEY(ATH_NUM) REFERENCES ATHLETES (ATH_NUM), FOREIGN KEY(DISCIPLINE_CODE) REFERENCES
DISCIPLINE (DISCIPLINE_CODE)
ON UPDATE CASCADE ON DELETE CASCADE);

COPY MEDALS (ATH_NUM, ATHLETE_NAME, ATHLETE_SHORT_NAME, MEDAL_TYPE, MEDAL, ATHLETE_SEX,
SPORT_EVENT, COUNTRY, DISCIPLINE_CODE)

FROM 'D:/SQL PROJECT/OLYMPICS WINTER GAMES/MEDALS.CSV'

DELIMITER ','

CSV HEADER;

SELECT * FROM MEDALS;

	Data Output	Explain	Messages	Notifications
	discipline_code [PK] character varying (10)	discipline character varying (200)	females character varying (10)	males character varying (10)
1	ALP	Alpine Skiing	153	157
2	BTH	Biathlon	106	107
3	BOB	Bobsleigh	46	119
4	CCS	Cross-Country Skiing	149	150
5	CUR	Curling	59	55
6	FSK	Figure Skating	74	74
7	FRS	Freestyle Skiing	127	141
8	IHO	Ice Hockey	236	311

Insights- Above snip is just for one table(DISCIPLINE) that was imported. The “CREATE” query function is used to create table by defining different column names as per the data-types. Then the rows are imported using “COPY” function from an external file which is places in the file-path mentioned in the fourth line of the query. Then “SELECT” function is used to show the imported data. This “COPY” function made the importing faster, data containing thousands of rows can be imported using just a single line query, rather than writing the “insert values” query numerous times. All the 5 tables were imported using the same method.

2- Identifying duplicates from “Athletes” dataset.

Code- SELECT ATH_NAME, COUNT(*) AS RECORDS

FROM ATHLETES

GROUP BY ATH_NAME

HAVING COUNT(*) > 1;

Data Output	Explain	Messages	Notifications
	ath_name character varying (200)	records bigint	
1	LEDECKA Ester	2	
2	BENDIKA Baiba	2	
3	ZHAO Jiawen	2	

Insights- The above query is to identify any duplicate values/rows in the dataset. It is totally up to the end-users which rows/values they want to keep unique for analysis.

3- Removing duplicates from the “Athletes” dataset.

Code- DELETE FROM ATHLETES

WHERE ATH_NUM IN (

SELECT ATH_NUM

FROM (SELECT ATH_NUM,

ROW_NUMBER() OVER (PARTITION BY ATH_NAME) AS ROWNUM

FROM ATHLETES) AS SUB

WHERE ROWNUM> 1);

Data Output	Explain	Messages	Notifications				
<div><div></div><div>ath_num</div><div>[PK] integer</div></div>	<div><div></div><div>ath_name</div><div>character varying (200)</div></div>	<div><div></div><div>short_name</div><div>character varying (100)</div></div>	<div><div></div><div>gender</div><div>character varying (10)</div></div>	<div><div></div><div>birth_date</div><div>date</div></div>	<div><div></div><div>birth_place</div><div>character varying (100)</div></div>	<div><div></div><div>birth_country</div><div>character varying (10)</div></div>	
1	1	AAGAARD Mikkel	AAGAARD M	Male	1995-10-18	FREDERIKSHAVN	Denmark
2	2	AALTO Antti	AALTO A	Male	1995-04-02	KITEE	Finland
3	3	AALTONEN Miro	AALTONEN M	Male	1993-06-07	JOENSUU	Finland
4	4	ABDELKADER Justin	ABDELKADER J	Male	1987-02-25	MUSKEGON, MI	United States of An
5	5	ABDI Fayik	ABDI F	Male	1997-10-07	SAN DIEGO, CA	United States of An
6	6	ABDUL-SABOOR Hakeem	ABDUL-SABOOR H	Male	1987-11-07	EAST ORANGE, NJ	United States of An
7	7	ABEDA Shannon	ABEDA S	Male	1996-05-15	FORT MCMURRAY, AB	Canada
8	8	ABELTSHAUSER Konrad	ABELTSHAUSER K	Male	✔ Successfully run. Total query runtime: 167 msec. 2894 rows affected.		

✓ Successfully run. Total query runtime: 167 msec. 2894 rows affected.

Insights- Initially, the dataset “athletes” was having 2897 rows in total, but after removing the duplicate athletes’ names it can be seen at the right-bottom corner of the screenshot that the total affected rows are 2894, which means the above-mentioned query has successfully removed all the duplicate rows, corresponding to athlete names.

4- Display the athletes’ names and their country’s names for all gold medal winners.






Code- SELECT A.ATH_NUM, A.ATH_FIRST_NAME, A.ATH_LAST_NAME , A.COUNTRY, M.MEDAL_TYPE

FROM ATHLETES A, MEDALS M

WHERE A.ATH_NUM = M.ATH_NUM

AND M.MEDAL_TYPE LIKE '%Gold%'

ORDER BY COUNTRY;

Data Output		Explain	Messages	Notifications	
	 ath_num integer	 ath_first_name character varying (200)	 ath_last_name character varying (200)	 country character varying (100)	 medal_type character varying (50)
1	1931	PARKER	Kathryn	Australia	Gold
2	2775	WRIGHT	Casey	Australia	Gold
3	61	ANTHONY	Jakara	Australia	Gold
4	2433	STROLZ	Johannes	Austria	Gold
5	2404	STEPAN	Sascha	Austria	Gold
6	828	GREIDERER	Lukas	Austria	Gold
7	1411	LADNER	Marco	Austria	Gold
8	567	EDER	Simon	Austria	Gold
9	652	FETTNER	Manuel	Austria	Gold
10	1872	ODOR	Gabriel	Austria	Gold
11	482	DERUGO	Anna	Belarus	Gold
12	862	GUSTIK	Maksim	Belarus	Gold

Insights- Here, the query is displaying the athletes who won the gold medal in the Olympics and their countries. The result was extracted using INNER JOIN between Athletes and Medals tables.

5- Calculate the total athletes based on different disciplines.

Code- `SELECT DISCIPLINE_CODE, COUNT(DISTINCT(ATH_NAME))`
`FROM ATHLETES`
`GROUP BY DISCIPLINE_CODE`
`ORDER BY DISCIPLINE_CODE`

Data Output	Explain	Messages	Notifications
	discipline_code character varying (10)	count bigint	
1	ALP	309	
2	BOB	165	
3	BTH	213	
4	CCS	298	
5	CUR	114	
6	FRS	267	
7	FSK	148	
8	IHO	547	
9	LUC	106	

Insights- Here, the discipline-wise number of athletes is shown by grouping them together using the COUNT aggregate function.

6- Create a new column that assigns a value based on the Heights of athletes:

- If the number of medals is greater than 1.8 meters , assign Tall.
- If the number of medals is between 1.6 to 1.79 meters, assign Medium Height.
- If the number of medals is below 1.59, assign Short.

Code- `SELECT ATH_NUM, ATH_FIRST_NAME, ATH_LAST_NAME,`
`CASE WHEN HEIGHT_M > 1.8 THEN 'TALL'`
`WHEN HEIGHT_M BETWEEN 1.6 AND 1.79 THEN 'MEDIUM HEIGHT'`

ELSE 'SHORT'

END AS COMPARISON

FROM ATHLETES

	ath_num [PK] integer	ath_first_name character varying (200)	ath_last_name character varying (200)	comparison text
1	1	AAGAARD	Mikkell	TALL
2	2	AALTO	Antti	SHORT
3	3	AALTONEN	Miro	SHORT
4	4	ABDELKADER	Justin	TALL
5	5	ABDI	Fayik	SHORT
6	6	ABDUL-SABOOR	Hakeem	SHORT
7	7	ABEDA	Shannon	SHORT
8	8	ABELTSHAUSER	Konrad	TALL
9	9	ABOLS	Rodrigo	TALL
10	10	ABRAMENKO	Oleksandr	SHORT
11	11	ABRUZZESE	Nick	MEDIUM HEIGHT
12	12	ACHEREEKTE	Carlijn	MEDIUM HEIGHT

Insights- Here the heights are segregated as Tall, Medium, Short as per the defined criteria, using WHEN CASE functions.

7- How many Golds, Silver, Bronze medals are there in total?

Code- *SELECT MEDAL_TYPE, MEDAL,*
SUM (MEDAL) AS MEDALS_TOTAL
FROM MEDALS
GROUP BY MEDAL_TYPE, MEDAL
ORDER BY MEDAL ASC

	medal_type character varying (50)	medal integer	medals_total bigint
1	Gold	1	230
2	Silver	2	458
3	Bronze	3	699

Insights- Here the code is written to show the total number of Gold, Silver and Bronze medals in the Medals table using SUM aggregation function with GROUP BY.

8- Display the coach details of gold medal winners.

Code- *SELECT COACH_NAME, COUNTRY, DISCIPLINE_CODE*
FROM COACHES
WHERE DISCIPLINE_CODE IN
(SELECT A.DISCIPLINE_CODE
FROM ATHLETES A, MEDALS M
WHERE A.ATH_NUM = M.ATH_NUM AND M.MEDAL_TYPE LIKE '%GOLD%')

	Data Output	Explain	Messages	Notifications
	coach_name character varying (100)	country character varying (100)	discipline_code character varying (10)	
1	BARES Jakub	Czech Republic	CUR	
2	BEIGHTON Sean	United States of America	CUR	
3	BEIGHTON Sean	United States of America	CUR	
4	BELANOV Sergei	ROC	CUR	
5	BOBARIKO Yevgeni	ROC	IHO	
6	CARDONE Andrea	Italy	CUR	
7	CARLSON Jesper	Denmark	CUR	
8	CARLSON Jesper	Denmark	IHO	
9	CHARETTE Pierre	Switzerland	CUR	

Insights – This query will tell the details of gold medal winners’ coaches because their win is directly related to their coaches, which can be helpful in the future when selecting the coach. The code is written using the nested query.

9- Display the information of the location of the event where the athletes won a maximum 3 or more gold medals.

Code- WITH NEW_TABLE AS

```
(SELECT ATH_NUM, ATHLETE_NAME, MAX(MEDAL) AS MAX_MEDALS, EVENT_LOCATIO
FROM MEDALS A INNER JOIN EVENTS E
ON A.DICIPLINE_CODE = E.DICIPLINE_CODE
GROUP BY ATH_NUM, ATHLETE_NAME, EVENT_LOCATION)
SELECT * FROM NEW_TABLE
WHERE MAX_MEDALS >= 3
```

	Data Output	Explain	Messages	Notifications
	ath_num integer	athlete_name character varying (200)	max_medals integer	event_location character varying (200)
1	905	HARLAUT Henrik	3	Big Air Shougang
2	1102	JASZAPATI Petra	3	Capital Indoor Stadium
3	2369	SOMMER Ryan	3	Yanqing National Sliding Centre
4	2433	STUPAK Yulia	3	Zhangjiakou National Cross-Country Skiing Centre
5	1593	MAIER Daniela	3	Big Air Shougang
6	1073	IVANOVA Tatyana	3	After Run 1
7	1268	KLAEBE Johannes Hoesflot	3	Zhangjiakou National Cross-Country Skiing Centre
8	1532	LIU Shaolin Sandor	3	Capital Indoor Stadium
9	1183	KARJAI AINFEN Sini	3	Wukesong Sports Centre

Insights- Here all the locations can be seen where athletes won a maximum 3 or more Gold Medals, this is obtained by using WITH function to create a view.

10- Create a view to display details of athletes participating in the doubles event and winning the gold medal.

Code- CREATE OR REPLACE VIEW DOUBLES_EVENTS AS

```
SELECT A.ATH_NUM, A.ATH_FIRST_NAME,A.ATH_LAST_NAME , A.COUNTRY, M.MEDAL_TYPE,
M.ATHLETE_SEX







FROM ATHLETES A INNER JOIN MEDALS M

ON A.ATH_NUM = M.ATH_NUM

WHERE M.ATHLETE_SEX IN ('O', 'X')

AND M.MEDAL_TYPE LIKE 'Gold';

SELECT * FROM DOUBLES_EVENTS
```

Data Output		Explain	Messages	Notifications		
	ath_num integer	 ath_first_name character varying (200)	 ath_last_name character varying (200)	 country character varying (100)	 medal_type character varying (50)	 athlete_sex character varying (50)
1	74	ARLT	Tobias	Germany	Gold	X
2	74	ARLT	Tobias	Germany	Gold	O
3	74	ARLT	Tobias	Germany	Gold	O
4	136	BAUER	Candy	Germany	Gold	O
5	141	BAUMGARTNER	Nick	United States of America	Gold	X
6	205	BOCH	Margot	France	Gold	X
7	206	BOE	Johannes Thingnes	Norway	Gold	X
8	212	BOESIGER	Jonas	Switzerland	Gold	X
9	248	BRENNAN	Rosie	United States of America	Gold	X
10	309	CAKMAKLI	Sabrina	Germany	Gold	X
11	378	CIRENZHANDUI	[null]	People's Republic of China	Gold	X
12	400	CONSTANTIN	Maria Ioana	Romania	Gold	X

Insight- - Views are generally used when data is to be accessed infrequently and data in the table get updated on a frequent basis. Here the view is created for end-users who can access only a selected part of the data set. The view is created using CREATE VIEW function to display all the athletes who participated in the doubles event and won gold medals. The filter was applied on “Athlete_sex” column because it segregates the single events from doubles.

11- Display country-wise total medals for all the sport events, as well as the grand total.

Code- SELECT SPORT_EVENT, COUNTRY, SUM(MEDAL) AS TOTAL_MEDALS

```
FROM MEDALS

GROUP BY CUBE(SPORT_EVENT, COUNTRY)

ORDER BY SPORT_EVENT, COUNTRY
```

Data Output	Explain	Messages	Notifications	Data Output	Explain	Messages	Notifications
▲	sport_event	country	total_medals	▲	sport_event	country	total_medals
character varying (200)	character varying (100)	bigint	character varying (200)	character varying (100)	bigint		
1	2-man	Germany	12	408	[null]	People's Republic of China	44
2	2-man	[null]	12	409	[null]	Poland	3
3	2-woman	Germany	6	410	[null]	Republic of Korea	32
4	2-woman	United States of America	6	411	[null]	ROC	178
5	2-woman	[null]	12	412	[null]	Slovakia	76
6	4-man	Canada	12	413	[null]	Slovenia	23
7	4-man	Germany	12	414	[null]	Spain	2
8	4-man	[null]	24	415	[null]	Sweden	66
9	Doubles	Austria	6	416	[null]	Switzerland	26
10	Doubles	Germany	3	417	[null]	Ukraine	2
11	Doubles	[null]	9	418	[null]	United States of America	127
12	Ice Dance	France	2	419	[null]	[null]	1387

Insight- Here total number of medals is displayed for each country, followed by the grand total of the medals in the end. The subtotal was done using the GROUP BY's CUBE function.

12- Display the youngest players as per their disciplines. Acquire the data only for those, whose birth date is given in the data.











Code- `SELECT *`

`FROM (SELECT *, RANK() OVER (PARTITION BY DISCIPLINE_CODE ORDER BY BIRTH_DATE DESC) AS BIRTH_RANK`

`FROM ATHLETES) AS NEW_T`

`WHERE BIRTH_RANK = 1`

`AND BIRTH_DATE IS NOT NULL`

Data Output		Explain	Messages	Notifications						
	 ath_num [PK] integer	 ath_first_name character varying (200)	 ath_last_name character varying (200)	 short_name character varying (11)	 gender character varying (10)	 birth_date date	 birth_place character varying (100)	 birth_country character varying (100)	 country character varying (10)	 discipline_code character varying (10)
1	2503	TEN	RAA	TEN RAA G	Female	2005-05-11	LUXEMBOURG	Luxembourg	Luxembourg	ALP
2	336	CERNANSKA	Viktoria	CERNANSKA V	Female	2002-03-29	BRATISLAVA	Slovakia	Slovakia	BOB
3	1409	LACHAT	Pablo	LACHAT P	Male	2000-10-20	MORGES	Switzerland	Switzerland	CUR
4	785	GLAZKOVA	Alexandra	GLAZKOVA A	Female	2006-03-15	MIASS	Russian Federation	ROC	FRS
5	405	CORLESS	Natalie	CORLESS N	Female	2003-11-27	NEW WESTMINSTER, BC	Canada	Canada	LUG
6	611	FAN	Haibin	FAN H	Male	2005-01-06	[null]	People's Republic of C...	People's Repu...	NCB
7	1465	LEE	Chaeun	LEE C	Male	2006-04-11	SUWON	Republic of Korea	Republic of K...	SBD
8	1059	INDRACKOVA	Anezka	INDRACKOVA A	Female	2006-07-30	JABLONEC NAD NISOU	Czech Republic	Czech Republic	SJP
9	2858	ZHAO	Dan	ZHAO D	Female	2002-12-14	[null]	People's Republic of C...	People's Repu...	SKN
10	1468	LEE	Eunice	LEE E	Female	2004-11-11	[null]	Republic of Korea	United States ...	STK

Insights- The code is written using the PARTITION BY function to display the youngest player based on their disciplines. The RANK = 1, will show the players who were born the latest as far as the dataset is concerned.

names

13- Show athletes' full name from athletes' table and the total number of medals they won.

Code- `SELECT CONCAT(ATH_FIRST_NAME , ' ', ATH_LAST_NAME) AS ATH_NAME, A.DISCIPLINE_CODE, SUM(M.MEDAL) AS TOTAL_MEDALS`

`FROM ATHLETES A`

`INNER JOIN MEDALS M`

ON A.ATH_NUM = M.ATH_NUM

GROUP BY ATH_NAME, A.DISCIPLINE_CODE

ORDER BY ATH_NAME

	ath_name text	discipline_code character varying (10)	total_medals bigint
1	AALTONEN Miro	IHO	1
2	ABRAMENKO Oleksandr	FRS	2
3	AICHER Emma	ALP	2
4	ALDOSHKIN Daniil	SSK	2
5	ALTHAUS Katharina	SJP	2
6	AMBROSE Erin	IHO	1
7	ANDERSEN Espen	NCB	1
8	ANDERSSON Ebba	CCS	3
9	ANDRONOV Sergei	IHO	2
10	ANTHONY Jakara	FRS	1
11	ANTTILA Marko	IHO	1
12	APARJODS Kristers	LUG	3

Insight- To show the athletes' Full Names, the first name and last name columns were concatenated using CONCAT function. Moreover, to show the total number of medals they won, the athletes and medals tables were joined using INNER JOIN function.

14- EVENT-WISE GOLD, SILVER & BRONZE MEDALS.

Code- `SELECT SPORT_EVENT, MEDAL_TYPE,`
`SUM (MEDAL) AS MEDALS_TOTAL`
`FROM MEDALS`
`GROUP BY ROLLUP(SPORT_EVENT,MEDAL_TYPE)`
`ORDER BY SPORT_EVENT ASC`

	sport_event character varying (200)	medal_type character varying (50)	medals_total bigint
1	2-man	Bronze	6
2	2-man	Gold	2
3	2-man	Silver	4
4	2-man	[null]	12
5	2-woman	Silver	4
6	2-woman	Bronze	6
7	2-woman	[null]	12
8	2-woman	Gold	2
9	4-man	Gold	4
10	4-man	Bronze	12
11	4-man	[null]	24
12	4-man	Silver	8

Insights- Here the code is displaying a total number of Gold, Silver, and Bronze Medals, for each event. The display is achieved by using ROLLUP function.

15- Create a materialized view for singles' events.

Code- `CREATE MATERIALIZED VIEW SINGLES_EVENT AS`

```
SELECT A.ATH_NUM, A.ATH_FIRST_NAME, A.COUNTRY, M.MEDAL_TYPE, M.ATHLETE_SEX
FROM ATHLETES A INNER JOIN MEDALS M
ON A.ATH_NUM = M.ATH_NUM
WHERE M.ATHLETE_SEX IN ('M', 'W');

SELECT * FROM SINGLES_EVENT
```

Data Output		Explain	Messages	Notifications	
	ath_num integer	ath_first_name character varying (200)	country character varying (100)	medal_type character varying (50)	athlete_sex character varying (50)
1	3	AALTONEN	Finland	Gold	M
2	10	ABRAMENKO	Ukraine	Silver	M
3	29	ALDOSHKIN	ROC	Silver	M
4	36	ALTHAUS	Germany	Silver	W
5	37	AMBROSE	Canada	Gold	W
6	43	ANDERSEN	Norway	Gold	M
7	47	ANDERSSON	Sweden	Bronze	W
8	56	ANDRONOV	ROC	Silver	M
9	61	ANTHONY	Australia	Gold	W
10	65	ANTTILA	Finland	Gold	M
11	120	BARNES	United States of America	Silver	W
12	137	BAUER	Germany	Silver	M

Insights- On the other hand Materialized Views are used when data is to be accessed frequently and data in table does not get updated on frequent basis. This code is showing all the athletes who participated in the singles' events.

16- WRITE A NESTED QUERY FOR INFORMATION ON ALL THE FEMALE ATHLETES WHO WON MEDALS FOR CANADA IN ICE HOCKEY.

Code- `SELECT ATH_NUM, ATH_FIRST_NAME, ATH_LAST_NAME, GENDER, COUNTRY, DISCIPLINE_CODE`

`FROM ATHLETES`







`WHERE GENDER LIKE 'Female'`

`AND ATH_NUM IN(`

`SELECT ATH_NUM`

`FROM MEDALS`

`WHERE COUNTRY LIKE 'Canada' AND DISCIPLINE_CODE LIKE 'IHO')`

	Data Output	Explain	Messages	Notifications	
	 ath_num [PK] integer 	ath_first_name character varying (200) 	ath_last_name character varying (200) 	gender character varying (10) 	country character varying (100) 
1	37	AMBROSE	Erin	Female	Canada
2	151	BELL	Ashton	Female	Canada
3	311	CAMERANESI	Dani	Female	United States of America
4	482	DERUGO	Anna	Female	Belarus
5	655	FIALKOVA	Paulina	Female	Slovakia
6	1859	NULLMEYER	Ali	Female	Canada
7	2086	RAST	Camille	Female	Switzerland

Insights- This query is displaying all the female athletes who won medals for Canada in Ice Hockey. The code is written using the nested query, by combining two tables' Athletes and Medals. Moreover, filters are applied using the WHERE condition.

References

- 1- https://www.kaggle.com/datasets/piterfm/beijing-2022-olympics?select=entries_discipline.csv
- 2- <https://olympics.com/en/olympic-games/beijing-2022>
- 3- https://en.wikipedia.org/wiki/2022_Winter_Olympics