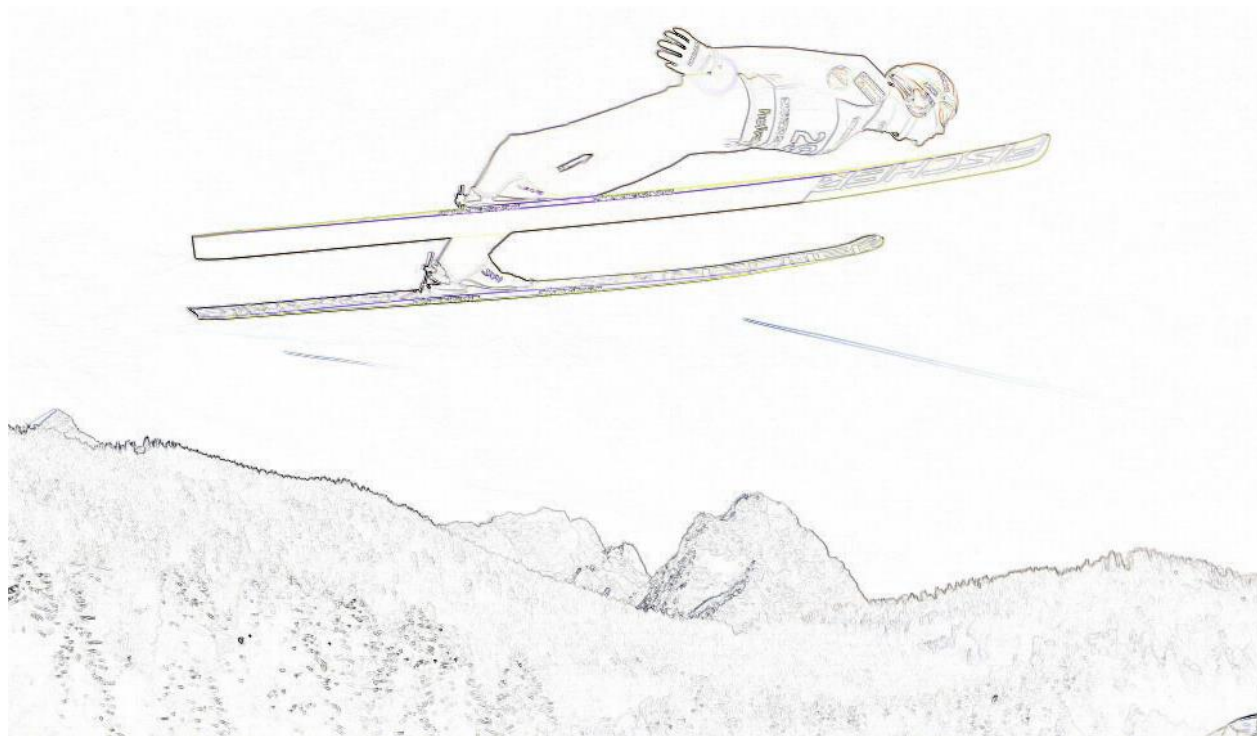


Ski Jumping Hill Records Database

Pawel Paszki (Applied Computing Year 2)

Student number: 20063617

Database Design



Contents

I.	Introduction.....	3
II.	Organisation description	3
i.	Info about organisation	3
ii.	Operation of organisation as a part of a database model	3
III.	EER Model.....	5
IV.	Relationships	6
V.	Tables	8
VI.	Queries	15
VII.	Security.....	16
VIII.	Conclusion	16
IX.	Bibliography:.....	17

I. Introduction

This report is about the database, which is to be implemented in the latter stage. The purpose of this report is to describe the database and why is the database implemented in the first place. Description includes the summary about the organisation, which will be responsible for maintaining the database. ER Model along with tables and description of each of the table and its content is also provided. List of typical queries (which may be slightly changed and/or expanded) is present in the report as well. The database is going to facilitate different types of access for regular and more privileged users. Conclusion section describes the next stage of implementation of the database.

II. Organisation description

i. Info about organisation

International Ski Federation (FIS) is the main governing body for several major winter sports. It is responsible for establishing and modifying rules regulating all governed disciplines, organising and supervising competitions, including delegating referees, etc. One of FIS's governed sport disciplines is Ski Jumping. (en.wikipedia.org (2016), fis-ski.com (2016))

Implementation of the database and querying its tables will give FIS a powerful analysis tool. The final goal of the implementation of the database is to gather information about all current and retired Ski Jumpers, their results obtained on all Hills. Database is also going to store information about the countries Ski Jumpers are/ were representing and complete history of coaches hired by those countries. Analysing the data will help to generate statistical data, which then may be used to improve functioning of the organisation.

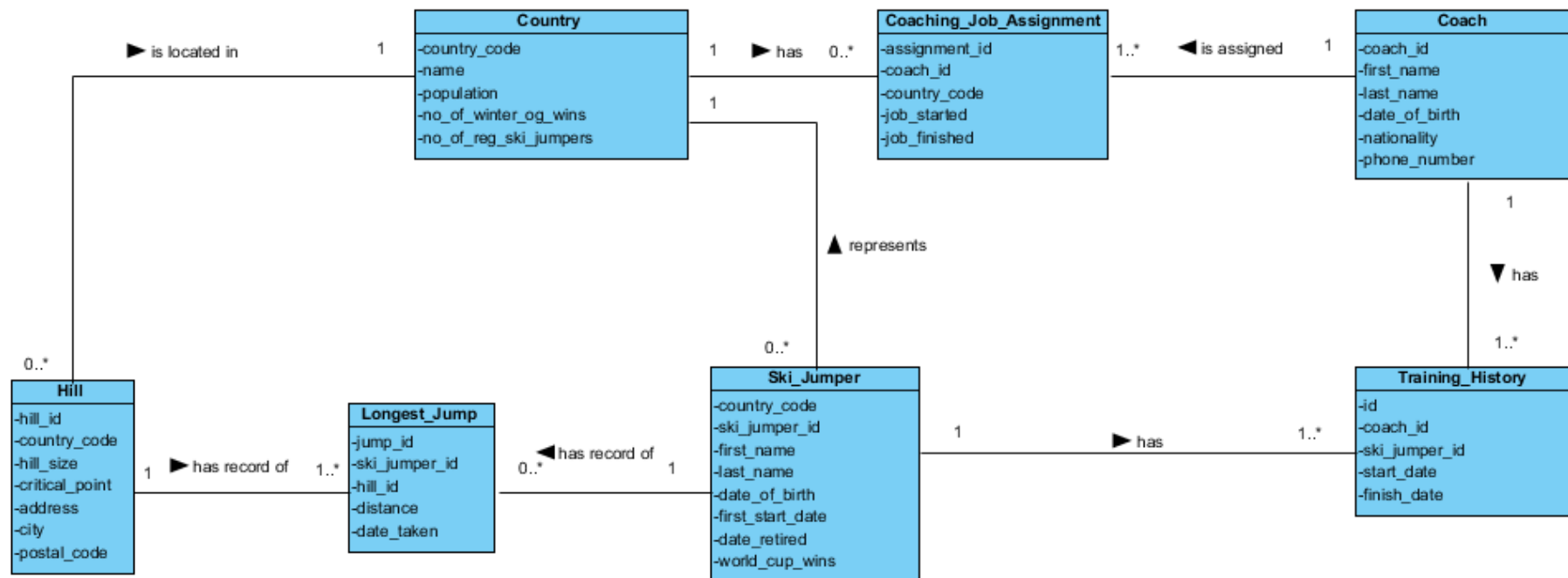
ii. Operation of organisation as a part of a database model

Each Ski Jumper represents only one country. One Country can be represented by zero or more Ski Jumpers. Country can hire Coaches. Only one Coach can have job assignment for a Country at any given time - Coach can work for other Country, once the job assignment for

previous Country has finished. Country can hire another Coach, while not currently hiring any. Coach and Ski Jumper have Training History – which defines the period of time, in which a given Ski Jumper and Coach work together (ie Ski Jumper is trained by Coach). There can be zero or more Hills located in a Country. Each Hill has a record of Longest Jumps. there is zero or one Longest Jump for each Ski Jumper on a given Hill recorded in the database.

Country code is stored along with Country name, population, number of winter Olympic Games wins and number of registered Ski Jumpers. For each Ski Jumper there is Country code and Ski Jumper id stored along with Ski Jumper's first and last name, date of birth, first World Cup start date, date retired (which can be null, if the competitor is still taking part in WC) and World Cup wins. Ski Jumper may or may not have taken a jump on a given Hill. If there was a record of a jump - it is stored in the Longest Jump table, which includes jump id, Ski Jumper id, Hill id, distance and date taken. For each Hill, its id, Country code, address, city, postal code, Hill size and critical point are recorded. Coaching Job Assignment table will store assignment id, Country code, Coach id and start / finish dates for the assignment. First name, last name, date of birth, nationality, and phone number along with Coach id will be recorded for each Coach. For the Training History start and finish date are stored along with id (primary key) and two foreign keys used to map related tables (Coach and Ski Jumper).

III. EER Model



IV. Relationships

Note:

Before the model took its shape shown in the diagram above there were two many-to-many relationships:

- Coach trains Ski_Jumper (1..* - 1..*). In order to resolve this relationship, Training_History entity was added
- Coach works for Country (1..* - 1..*). In order to resolve this relationship, Coaching_Job_Assignment entity was introduced

Relationships and their primary / foreign key relations:

- Hill is located in Country:

Multiplicity: 0..* - 1 (there can be zero or more Hills in a given country. Each Hill can be located in only one Country). Mapped by country_code (primary key in Country table), which becomes a foreign key in Hill table.

- Hill has record of Longest_Jump:

Multiplicity: 1 – 1..* (for each Hill there is at least one Longest_Jump. Any particular Longest_Jump is recorded for one Hill). Mapped by hill_id (primary key in Hill table), which is a foreign key in Longest_Jump table

- Ski_Jumper has record of Longest_Jump:

Multiplicity: 1 – 0..* (for each Ski_Jumper can be record of a Longest_Jump. Any particular Longest_Jump is recorded for one Ski_Jumper). Mapped by ski_jumper_id (primary key in Ski_Jumper table), which is a foreign key in Longest_Jump table

- Ski_Jumper represents Country:

Multiplicity: 0..* – 1 (Country can be represented by zero or more Ski_Jumpers, but each Ski_Jumper can only represent one Country).

Mapped by country_code (primary key in Country table), which is a foreign key in Ski_Jumper table

- Country has Coaching_Job_Assignment:

Multiplicity: 1 – 0..* (Country can have zero or more Coaching Job Assignment, but each Coaching Job Assignment can only be created for one Country). Mapped by country_code (primary key in Country table), which is a foreign key in Coaching_Job_Assignment table

- Coach is assigned Coaching_Job_Assignment:

Multiplicity: 1 – 1..* (Coach can have one or more Coaching Job Assignment, but each Coaching Job Assignment can only be created for one Coach). Mapped by coach_id (primary key in Coach table), which is a foreign key in Coaching_Job_Assignment table

- Ski_Jumper has Training_History:

Multiplicity: 1 – 1..* (one Ski_Jumper can have one or more records of Training_History (in this model training history means working with one Coach). Mapped by ski_jumper_id (primary key in Ski_Jumper table and foreign key in Training_History table)

- Coach has Training_History:

Multiplicity: 1 – 1..* (one Coach can have one or more records of Training_History. Mapped by coach_id (primary key in Coach table and foreign key in Training_History table)

V. Tables

Country						
Column name	Data type	Nullable	Default	PK / FK	Check	Description and example
country_code	CHAR(3)	no	no	primary key	no	represents officially assigned by ISO, internationally recognizable, country code. <u>examples: IRL (Ireland), GER (Germany)</u>
name	VARCHAR2(40)	no	no	no	no	name of a given country in English <u>Example: Germany</u>
population	NUMBER(9,0)	no	no	no	no	Estimated number of people currently living in given country <u>Example: 15,000,000</u>
no_of_winter_og_wins	NUMBER(3,0)	no	no	no	no	total number of wins in all winter Olympic Games by all competitors from given country <u>Example: 239</u>
no_of_reg_ski_jumpers	NUMBER(5,0)	no	no	no	no	number of all registered ski jumpers in all clubs at all levels in given country <u>Example: 4231</u>

Hill						
Column name	Data type	Nullable	Default	PK / FK	Check	Description and example
hill_id	NUMBER(3,0)	no	no	primary key	no	numeric value representing unique hill identifier. <u>Example: 10</u>
country_code	CHAR(3)	no	no	foreign key	no	represents officially assigned by ISO, internationally recognizable, country code. <u>examples: IRL (Ireland), GER (Germany)</u>
hill_size	NUMBER(3,0)	no	no	no	no	size of the hill – distance in meters from the end of the inrun. Once it is exceeded – during competition jury can decide to shorten the length of the inrun. <u>Example: 140</u>
critical_point	NUMBER(3,0)	no	no	no	yes hill_size > critical_point	distance from the end of inrun (in meters), which gives 60 points that constitute the overall jump note <u>Example: 125</u>
address	VARCHAR2(40)	yes	no	no	no	street number and name, on which the hill is located <u>Example: Kongeveien 5</u>
city	VARCHAR2(30)	no	no	no	no	the city in which the hill is located <u>Example: Oslo</u>
postal_code	VARCHAR2(20)	no	no	no	no	code used to uniquely identify a particular geographic area. <u>Example: 0001</u>

Ski_Jumper						
Column name	Data type	Nullable	Default	PK / FK	Check	Description and example
ski_jumper_id	NUMBER(6,0)	no	no	primary key	no	numeric value representing unique ski jumper identifier. <u>Example: 100</u>
country_code	CHAR(3)	no	no	foreign key	no	represents officially assigned by ISO country code. <u>examples: IRL (Ireland), GER (Germany)</u>
first_name	VARCHAR2(25)	no	no	no	no	ski jumpers' first name taken from his birth certificate <u>Example: Matti</u>
last_name	VARCHAR2(25)	no	no	no	no	ski jumpers' last name taken from his birth certificate <u>Example: Hautamaeki</u>
date_of_birth	DATE	no	no	no	no	ski jumpers' date of birth taken from his birth certificate <u>Example: 20/10/1989</u>
first_start_date	DATE	no	no	no	yes first_start_date > date_of_birth	date of first official start in a world cup competition <u>Example: 20/10/2007</u>
date_retired	DATE	no	no	no	yes date_retired > first_start_date	date of last official start in a world cup competition before finishing career as a competitor <u>Example: 21/03/2015</u>
world_cup_wins	NUMBER(2,0)	no	0	no	no	Number of won World Cup competitions <u>Example: 12</u>

Longest_Jump						
Column name	Data type	Nullable	Default	PK / FK	Check	Description and example
jump_id	NUMBER(8,0)	no	no	primary key	no	numeric value representing unique jump identifier. <u>Example: 99</u>
ski_jumper_id	NUMBER(6,0)	no	no	foreign key	no	numeric value representing unique ski jumper identifier. <u>Example: 100</u>
hill_id	NUMBER(3,0)	no	no	foreign key	no	numeric value representing unique hill identifier. <u>Example: 10</u>
distance	NUMBER(4,1)	no	no	no	no	length of the attempt measured from the end of the inrun to the landing place (in meters) <u>Example: 123,5</u>
date_taken	DATE	no	no	no	no	date, when the recorded attempt was taken <u>Example: 04/01/2015</u>

Coach						
Column name	Data type	Nullable	Default	PK / FK	Check	Description and example
coach_id	NUMBER(3,0)	no	no	primary key	no	numeric value representing unique coach identifier. <u>Example: 10</u>
first_name	VARCHAR2(25)	no	no	no	no	coaches' first name taken from his birth certificate <u>Example: Lukasz</u>
last_name	VARCHAR2(25)	no	no	no	no	coaches' last name taken from his birth certificate <u>Example: Kruczek</u>
date_of_birth	DATE	no	no	no	no	coaches' date of birth taken from his birth certificate <u>Example: 20/10/1965</u>
nationality	VARCHAR2(30)	no	no	no	no	country of origin <u>Example: polish</u>
phone_number	VARCHAR2(20)	no	no	no	no	contact number <u>Example: 353 85 1234567</u>

Coaching_Job_Assignment						
Column name	Data type	Nullable	Default	PK / FK	Check	Description and example
assignment_id	NUMBER(4,0)	no	no	primary key	no	numeric value representing unique job assignment identifier. <u>Example: 10</u>
coach_id	NUMBER(3,0)	no	no	foreign key	no	numeric value representing unique coach identifier. <u>Example: 20</u>
country_code	CHAR(3)	no	no	foreign key	no	represents officially assigned by ISO country code. <u>examples: IRL (Ireland), GER (Germany)</u>
job_started	DATE	no	no	no	no	date, when a coach was hired and contract was signed <u>Example: 01/01/2004</u>
job_finished	DATE	no	no	no	no	date, when the job contract expired <u>Example: 23/03/2012</u>

Training_History						
Column name	Data type	Nullable	Default	PK / FK	Check	Description and example
id	NUMBER(6,0)	no	no	primary key	no	numeric value representing unique job assignment identifier. <u>Example: 10</u>
coach_id	NUMBER(3,0)	no	no	foreign key	no	numeric value representing unique coach identifier. <u>Example: 20</u>
ski_jumper_id	NUMBER(6,0)	no	no	foreign key	no	numeric value representing unique ski jumper identifier. <u>Example: 100</u>
start_date	DATE	no	no	no	no	date, when given coach and ski jumper started working together <u>Example: 01/01/2004</u>
finish_date	DATE	no	no	no	no	date, when given coach and ski jumper finished working together <u>Example: 23/03/2012</u>

VI. Queries

Note: the format in which the queried results will be displayed will be specified in the latter stage

These are the queries, which will be used to obtain specific information from the database once the database is implemented (the list is not final):

- All the hill records (ski jumper name, length of the jump, date and name and size of the hill will be printed)
- All hill records set by Ski_Jumper, who is not a citizen of a country, in which the hill is placed (location, critical_point, hill_size, first_name (Ski Jumper), last_name (Ski_Jumper), date_taken (Longest_Jump), distance (Longest_Jump))
- All ski jumpers from specified country (first_name, last_name, date_of_birth, world_cup_wins)
- All ski jumpers, who don't have a hill in their country and yet managed to won over 10 World Cup competitions (first_name, last_name, name (Country), world_cup_wins)
- All ski jumpers, which have all their longest jumps of length over 105% of the critical point of a hill (first_name (Ski Jumper), last_name (Ski_Jumper), date_taken (Longest_Jump), distance (Longest_Jump), critical_point (Hill))
- 5 ski jumpers with the most World Cup wins (first_name, last_name, name (Country), world_cup_wins)
- All ski jumpers with at least one longest jump greater than 200 m with the distance of that jump (first_name, last_name, name (Country), distance (Longest_Jump), date_taken (Longest_Jump))
- All ski jumpers, which have been trained by a coach for at least 5 years (first_name, last_name (Ski Jumper), first_name, last_name (Coach), duration (taken from Training_History table))
- Ski jumper with no official hill record and over 20 wins (first_name, last_name, name (Country), world_cup_wins)
- Ski jumper with country of population below 30 million and less than 3 hills, who won the most WCs (first_name, last_name, name (Country), hill count, world_cup_wins)
- Ski jumpers with no World Cup win from the countries, which have at least 200 winter Olympic Games medals (first_name, last_name, name (Country), number_of_winter_og_wins (Country))

- All hills with hill size over 200m (name (Country), city (Hill), hill_size (Hill), record (Longest_Jump))
- Coach, who worked for most countries (first_name, last_name (Coach), job_started (Coaching_Job_Assignment), job_finished (Coaching_Job_Assignment), name (Country))
- Coach, who worked for the longest period of time for one country and duration of the assignment (first_name, last_name (Coach), job_started (Coaching_Job_Assignment), job_finished (Coaching_Job_Assignment), name (Country))
- Country with the biggest number of hills (name (Country), number of hills)
- The shortest jump from the database, which is the hill record (distance (Longest_Jump), date_taken, (Longest_Jump), city (Hill), hill_size (Hill), first_name (Ski_Jumper), last_name (Ski_Jumper))

VII. Security

There will be security features implemented, such as different read/ write permissions for different kind of database users. After the database is implemented – implementation document will describe those features in greater detail.

VIII. Conclusion

By now the model for the database has been created and data types for all tables' columns have been determined and specified. Next stage will involve creating tables in the Oracle Application Express. Once the tables are created – data will be gathered and number of rows will be added to all the tables. The next stage's goal is to gather enough data to query it with the queries specified above and get meaningful results. Final goal is to gather all the data to make the database complete.

IX. Bibliography:

en.wikipedia.org (2016) Available at: https://en.wikipedia.org/wiki/International_Ski_Federation (accessed: 27th of February 2015)

fis.com (2016). Available at: <http://www.fis-ski.com/inside-fis/about/fis-structure/structure/index.html> (accessed 1st of March 2016)