GOLF CLUBS DB

Database Project

25 April 2014

Vishnu Meduri

Table of Contents

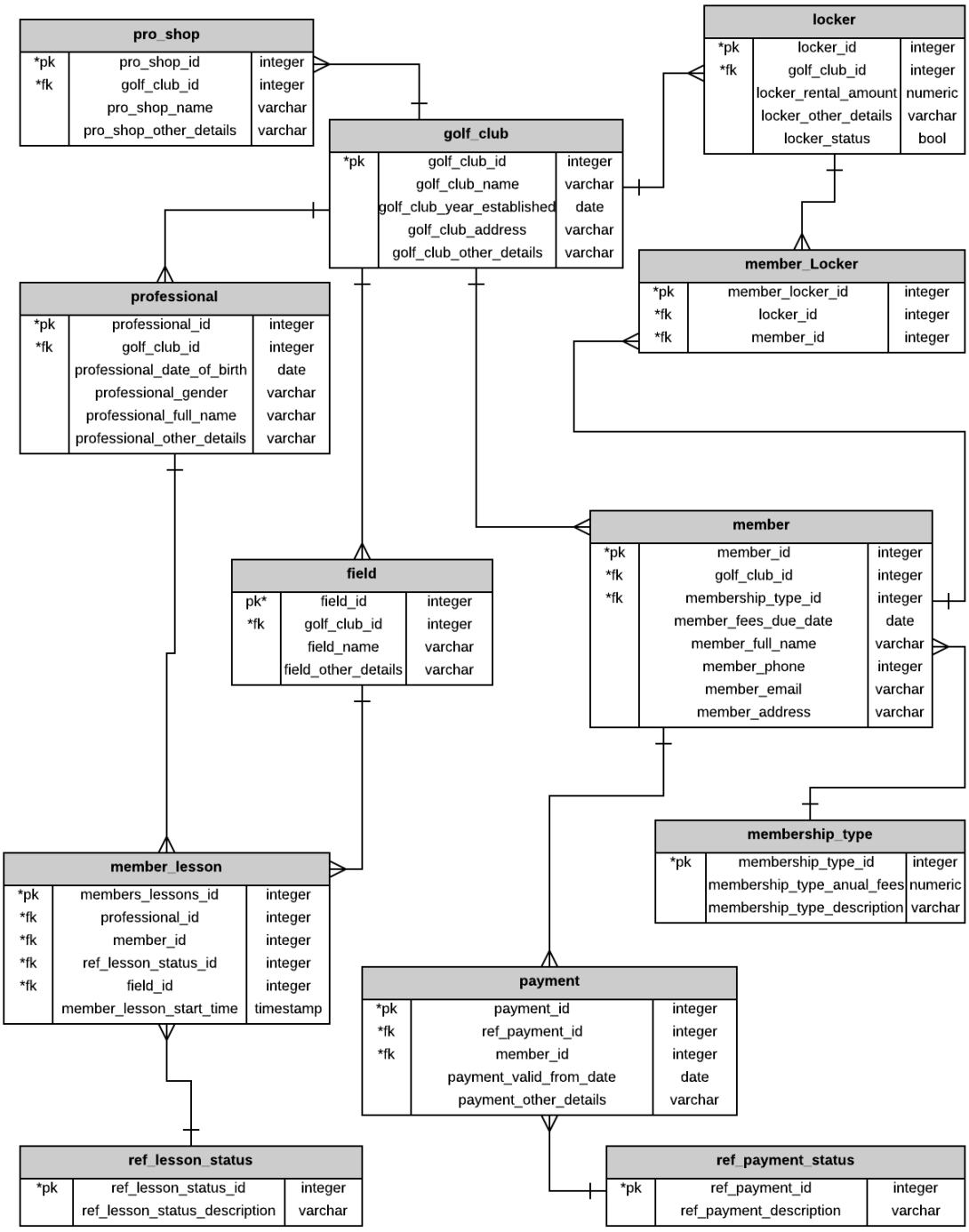
Summary

This Database records the information about different golf clubs and their members and the privileges of the members of a golf club. It gives specific information of the type of field of the golf club and other details of it. It also records information of individual members and their data.

It shows the professionals at a golf club and the lessons they provide in a particular golf club which is an interesting feature of the golf club DB.

The database is designed in a convenient way to add different golf clubs and their members or members for golf clubs.

ENTITY RELATIONSHIP DIAGRAM



Tables

Field Table

Purpose

It describes name of field and some comments about the fields.

Functional Dependencies

field_id → golf_club_id, field_name, field_other_details

Table Create Statements

```
CREATE TABLE field
(
  field_id integer NOT NULL,
  golf_club_id integer,
  field_name character varying(255),
  field_other_details character varying(255),
  CONSTRAINT pk_field_id PRIMARY KEY (field_id),
  CONSTRAINT field_golf_club_id_fkey FOREIGN KEY (golf_club_id)
      REFERENCES golf_club (golf_club_id)
)
```

Sample Data

	field_id integer	golf_club_id integer	field_name character varying(255)	field_other_details character varying(255)
1	32	2	jumper field	good field
2	33	3	cellular field	good field
3	34	4	our field	best field
4	35	5	exit field	
5	36	6	coca cola field	jumper field
6	37	7	pin field	good field
7	38	8	right field	
8	31	1	jose field	

Golf Club Table

Purpose

This tables gives the information about different golf clubs from all over the country.

Functional Dependencies

Golf_club_id → golf_club_name, golf_club_year_established, golf_club_address, golf_club_other_details

Table Create Statements

```
CREATE TABLE golf_club
(
  golf_club_id integer NOT NULL,
  golf_club_name character varying(255),
  golf_club_year_established date,
  golf_club_address character varying(255),
  golf_club_other_details character varying(255),
  CONSTRAINT pk_golf_club_id PRIMARY KEY (golf_club_id)
)
```

Sample Data

	golf_club_id integer	golf_club_name character varying(255)	golf_club_year_established date	golf_club_address character varying(255)	golf_club_other_details character varying(255)
1	1	jhons park	1991-01-03	new york	good club
2	2	mickey	1941-01-03	california	professional club
3	3	harry field	2001-12-05	san diego	child club
4	4	big field	2000-08-13	los angeles	club
5	5	right park	2004-08-13	san francisco	
6	6	left park	2007-03-14	new jersey	
7	7	mouse field	1990-09-15	houston	very good field
8	8	san jose field	1994-09-15	san jose	very good field

Locker Table

Purpose

This table gives the details about the locker systems available at golf clubs.

Functional Dependencies

Locker_id → golf_club_id, locker_rental_amount, locker_other_details, locker_available

Table Create Statements

```
CREATE TABLE locker
(
  locker_id integer NOT NULL,
  golf_club_id integer,
  locker_rental_amount numeric,
  locker_other_details character varying(255),
  locker_aviable boolean,
  CONSTRAINT pk_locker_id PRIMARY KEY (locker_id),
  CONSTRAINT locker_golf_club_id_fkey FOREIGN KEY (golf_club_id)
      REFERENCES golf_club (golf_club_id)
)
```

Sample Data

	locker_id integer	golf_club_id integer	locker_rental_amount numeric	locker_other_details character varying(255)	locker_aviable boolean
1	40	1	15	good locker	f
2	41	5	16	big locker	f
3	42	5	16	big locker	f
4	45	3	14	good locker	f
5	47	4	15	big locker	f
6	44	2	10	good locker	t
7	46	6	12	good locker	t
8	48	4	15	big locker	t
9	43	7	10	small locker	t

Member Table

Purpose

The information about people who belong to different golf clubs such as their golf club id, full name, address etc.

Functional Dependencies

Member_id → golf_club_id, membership_type_id, member_fees_due_date, member_full_name, member_phone , member_email, member_address, member_other_details

Table Create Statements

```
CREATE TABLE member
(
  member_id integer NOT NULL,
  golf_club_id integer,
  membership_type_id integer,
  member_fees_due_date date,
  member_full_name character varying(255),
  member_phone integer,
  member_email character varying(255),
  member_address character varying(255),
  member_other_details character varying(255),
  CONSTRAINT pk_member_id PRIMARY KEY (member_id),
  CONSTRAINT member_golf_club_id_fkey FOREIGN KEY (golf_club_id)
      REFERENCES golf_club (golf_club_id) ,
  CONSTRAINT member_membership_type_id_fkey FOREIGN KEY (membership_type_id)
      REFERENCES membership_type (membership_type_id)
)
```

Sample Data

	member_id integer	golf_club_id integer	membership_type_id integer	member_fees_due_date date	member_full_name character varying(255)	member_phone integer
1	8	1	1	2010-01-01	vanessa rain	0
2	7	2	2	2011-01-01	josh rass	0
3	6	3	3	2012-01-01	daniel one	0
4	5	4	4	2012-01-01	weak wanwright	0
5	4	5	5	2011-01-01	jen jensen	5555555
6	2	1	1	2011-01-01	cristina hernandez	5555555
7	1	1	1	2011-01-01	juan hernandez	5555555

Member Lesson Table

Purpose

Details about the lessons and the member taking the lesson.

Functional Dependencies

Member_lesson_id → professional_id, member_id, ref_lesson_status_id, field_id, member_lesson_start_time

Table Create Statements

```
CREATE TABLE member_lesson
(
  member_lesson_id integer NOT NULL,
  professional_id integer,
  member_id integer,
  ref_lesson_status_id integer,
  field_id integer,
  member_lesson_start_time timestamp with time zone,
  CONSTRAINT pk_member_lesson_id PRIMARY KEY (member_lesson_id),
  CONSTRAINT member_lesson_field_id_fkey FOREIGN KEY (field_id)
      REFERENCES field (field_id) ,
  CONSTRAINT member_lesson_member_id_fkey FOREIGN KEY (member_id)
      REFERENCES member (member_id) ,
  CONSTRAINT member_lesson_professional_id_fkey FOREIGN KEY (professional_id)
      REFERENCES professional (professional_id),
  CONSTRAINT member_lesson_ref_lesson_status_id_fkey FOREIGN KEY (ref_lesson_status_id)
      REFERENCES ref_lesson_status (ref_lesson_status_id)
)
```

Sample Data

member_lesson_id integer	professional_id integer	member_id integer	ref_lesson_status_id integer	field_id integer	member_lesson_start_time timestamp with time zone
1	21	1	1	31	2009-03-03 11:30:00-08
2	22	7	2	32	2009-02-02 11:30:00-08
3	23	6	1	33	2009-02-01 11:30:00-08
4	24	5	3	34	2009-01-09 13:00:00-08
5	25	4	2	35	2009-01-18 13:00:00-08
6	21	1	1	31	2009-03-03 14:00:00-08

Member Locker Table

Purpose

The tables gives the information about the locker of each member.

Functional Dependencies

Member_locker_id → locker_id, member_id

Table Create Statements

```
CREATE TABLE member_locker
(
  member_locker_id integer NOT NULL,
  locker_id integer,
  member_id integer,
  CONSTRAINT pk_member_locker_id PRIMARY KEY (member_locker_id),
  CONSTRAINT member_locker_locker_id_fkey FOREIGN KEY (locker_id)
      REFERENCES locker (locker_id),
  CONSTRAINT member_locker_member_id_fkey FOREIGN KEY (member_id)
      REFERENCES member (member_id)
)
```

Sample Data

member_locker_id integer	locker_id integer	member_id integer
31	40	8
32	41	4
33	42	4
35	40	2
34	47	5
37	40	1
36	45	6

Membership Type Table

Purpose

The type of membership and the fees and description of that.

Functional Dependencies

Membership_type_id → membership_type_anual_fees, membership_type_description

Table Create Statements

```
CREATE TABLE membership_type
(
  membership_type_id integer NOT NULL,
  membership_type_anual_fees numeric,
  membership_type_description character varying(255),
  CONSTRAINT pk_membership_type_code PRIMARY KEY (membership_type_id)
)
```

Sample Data

membership_type_id integer	membership_type_anual_fees numeric	membership_type_description character varying(255)
1	400	gold
2	300	silver
3	200	bronze
4	600	platinum
5	100	normal

Payment Table

Purpose

This table keeps the record of the payments made for memberships by the members.

Functional Dependencies

Payment_id → ref_payment_id, member_id, payment_valid_from_date, payment_other_details

Table Create Statements

```
CREATE TABLE payment
(
  payment_id integer NOT NULL,
  ref_payment_id integer,
  member_id integer,
  payment_valid_from_date date,
  payment_other_details character varying(255),
  CONSTRAINT pk_payment_id PRIMARY KEY (payment_id),
  CONSTRAINT payment_member_id_fkey FOREIGN KEY (member_id)
      REFERENCES member (member_id) ,
  CONSTRAINT payment_ref_payment_id_fkey FOREIGN KEY (ref_payment_id)
      REFERENCES ref_payment_status (ref_payment_id)
)
```

Sample Data

	payment_id integer	ref_payment_id integer	member_id integer	payment_valid_from_date date	payment_other_details character varying(255)
1	2	2	8	2009-03-01	complete
2	3	2	6	2009-05-04	cancelled
3	7	1	4	2009-07-02	
4	1	1	7	2009-02-01	incompleted
5	4	2	2	2009-10-01	completed
6	5	1	1	2009-12-01	completed
7	6	2	6	2008-12-05	completed

Pro Shop Table

Purpose

The table gives details about the shop from which professionals can buy.

Functional Dependencies

Pro_shop_id → golf_club_id, pro_shop_name, pro_shop_other_details

Table Create Statements

```
CREATE TABLE pro_shop
(
  pro_shop_id integer NOT NULL,
  golf_club_id integer,
  pro_shop_name character varying(255),
  pro_shop_other_details character varying(255),
  CONSTRAINT pk_pro_shop_id PRIMARY KEY (pro_shop_id),
  CONSTRAINT pro_shop_golf_club_id_fkey FOREIGN KEY (golf_club_id)
      REFERENCES golf_club (golf_club_id)
)
```

Sample Data

	pro_shop_id integer	golf_club_id integer	pro_shop_name character varying(255)	pro_shop_other_details character varying(255)
1	10	3	jhon	good shop
2	11	1	jennifer	good shop
3	12	4	riya	shop
4	13	6	michel	jump shop
5	14	7	texeira	very good shop
6	15	7	texeira II	very good shop
7	16	8	rain	very good shop
8	17	5	los4	
9	18	3	T.H.E	

Professional Table

Purpose

The professional according to their golf clubs are recorded in this table.

Functional Dependencies

Professional_id → golf_club_id, professional_date_of_birth, professional_gender, professional_full_name, professional_other_details

Table Create Statements

```
CREATE TABLE professional
(
  professional_id integer NOT NULL,
  golf_club_id integer,
  professional_date_of_birth date,
  professional_gender character varying(1),
  professional_full_name character varying(255),
  professional_other_details character varying(255),
  CONSTRAINT pk_pro_id PRIMARY KEY (professional_id),
  CONSTRAINT professional_golf_club_id_fkey FOREIGN KEY (golf_club_id)
      REFERENCES golf_club (golf_club_id)
)
```

Sample Data

	professional_id integer	golf_club_id integer	professional_date_of_birth date	professional_gender character varying(1)	professional_full_nam character varying(255
1	21	1	1991-01-01	m	jhon apple
2	22	2	1992-01-01	m	jose diaz
3	23	3	1980-01-01	m	david faitelson
4	24	4	1981-05-12	m	paul wright
5	25	5	1982-06-13	m	tiger woods
6	27	7	1989-08-19	m	shin shan
7	28	8	1984-11-20	m	prince
8	26	6	1988-08-16	f	lorena ochoa

Ref Lesson Status Table

Purpose

The status of the lesson whether complete, incomplete or cancelled.

Functional Dependencies

Ref_lesson_status_id → ref_lesson_status_description

Table Create Statements

```
CREATE TABLE ref_lesson_status
(
  ref_lesson_status_id integer NOT NULL,
  ref_lesson_status_description character varying(255),
  CONSTRAINT pk_lesson_status_cod PRIMARY KEY (ref_lesson_status_id)
)
```

Sample Data

	ref_lesson_status_id integer	ref_lesson_status_description character varying(255)
1	1	complete
2	2	incomplete
3	3	cancelled

Ref Payment Table Table

Purpose

This table describes the method of payment either credit or cash.

Functional Dependencies

Ref_payment_id → ref_payment_description

Table Create Statements

```
CREATE TABLE ref_payment_status
(
  ref_payment_id integer NOT NULL,
  ref_payment_description character varying(255),
  CONSTRAINT pk_payment_method_code PRIMARY KEY (ref_payment_id)
)
```

Sample Data

	ref_payment_id integer	ref_payment_description character varying(255)
1	1	credit
2	2	cash

Stored Procedures

List of Lockers Available

Purpose

It shows us the list of lockers available for the members.

Create Statements

```
CREATE OR REPLACE FUNCTION list_of_lockers_aviables()
  RETURNS SETOF list_of_lockers_aviables AS
$BODY$
   select 'Locker N°' || locker_id
   from locker
   where locker_aviable is true
   $BODY$
  LANGUAGE sql VOLATILE
```

Sample Data

	list_of_lockers_aviables list_of_lockers_aviables
1	("Locker N°44")
2	("Locker N°46")
3	("Locker N°48")
4	("Locker N°43")

Number of Jobs By Professional

Purpose

The number of professionals giving the lessons for members is given by this stored procedure.

Create Statements

```
CREATE OR REPLACE FUNCTION number_jobs_by_professional()
  RETURNS SETOF number_jobs_by_professional AS
$BODY$
   select ' Professional N°' || professional.professional_id ||' with: '
   count(professional.*) ||' jobs ' from professional, member_lesson
where professional.professional_id = member_lesson.professional_id
group by professional.professional_id
   $BODY$
  LANGUAGE sql VOLATILE
```

Sample Data

	number_jobs_by_professional number_jobs_by_professional
1	(" Professional N°23 with: 1 jobs ")
2	(" Professional N°25 with: 1 jobs ")
3	(" Professional N°22 with: 1 jobs ")
4	(" Professional N°21 with: 2 jobs ")
5	(" Professional N°24 with: 1 jobs ")

Triggers

Block Locker

Purpose

When a locker is created for a member it is blocked by this trigger.

Create Statements

```
CREATE OR REPLACE FUNCTION block_locker()
  RETURNS trigger AS
$BODY$
    BEGIN
        update locker set locker_aviable = FALSE where new.locker_id = locker_id;
        return new;
    END;
$BODY$
  LANGUAGE plpgsql VOLATILE
|
```

Unblock Locker

Purpose

When a member is removed from the golf club then the locker is unblocked by this trigger.

Functional Dependencies

```
CREATE OR REPLACE FUNCTION unlock_locker()
  RETURNS trigger AS
$BODY$
    BEGIN
        update locker set locker_aviable = TRUE where old.locker_id = locker_id;
        return old;
    END;
$BODY$
  LANGUAGE plpgsql VOLATILE
```

SECURITY

The DBA (Database Administrator) is one with all the permissions for the database. They can be assigned to him in the following way.

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO DBA;

KNOWN PROBLEMS & FUTURE ENHANCEMENTS

The database mainly collects the golf clubs but doesn't tell us about the owners of the club or the group of people behind its establishment. This can be added to database so that we know which golf club is owned by whom.

The golf club table itself does not include the fee for its member which is again stored in a different table and is an extra step to get that particular information of that golf club.

As a future enhancement we could add the table for information about the different championships conducted by the golf clubs. And should also include the trophies given by the golf clubs.

There can also be a rating for each member in the member table which can then determine the overall rating of the golf club.