

DATABASE MANAGEMENT SYSTEMS

PROJECT ASSIGNMENT

Lecturer: Gozde Yolcu Oztel

Name Surname	Student Number	Student E-mail
Merve KILCI	B211202375	merve.kilci@ogr.sakarya.edu.tr
Masa IBRAHIM AGHA	B201202554	masa.agha@ogr.sakarya.edu.tr

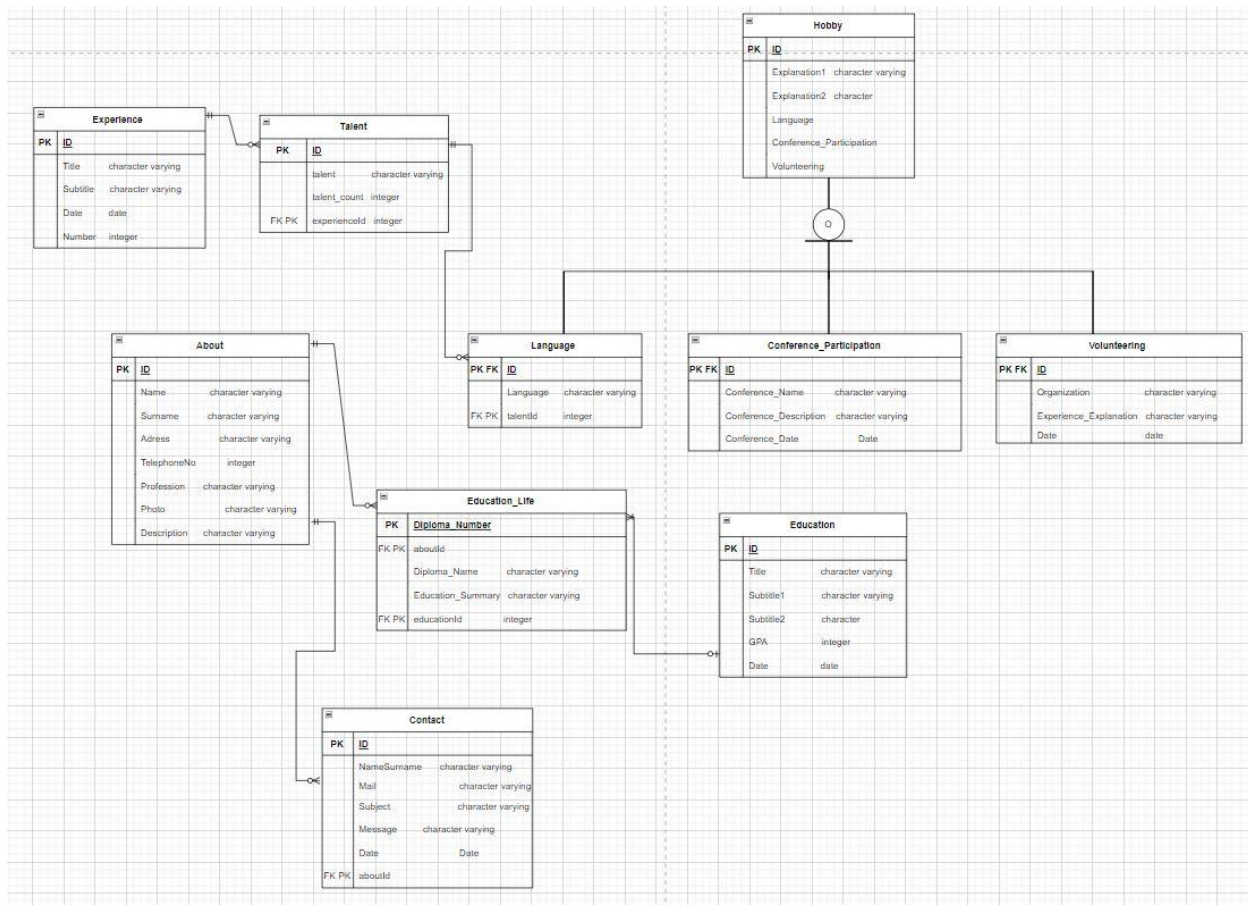
Introduction of the program

You don't know how to set up your own resume website, Simple! CVmaker is the helper with the best online resume builder. It will guide you through the resume preparation process all the way to your dream job. CVmaker will lead you to start building a resume website, with its simple efficient UI and UX design we created a menu that holds the main criteria that companies look for and the employee needs to show precisely.

Business Rules

1. The website user will have an Experience part that will hold Id, Title, Subtitle, Date, and experiences number.
2. The website will have a Talent section holding Id, talent, talent count, and the experience Id.
3. Experience might have 0 or many talents.
4. Each Talent can be gained in only one Experience.
5. The website has the Language part that holds the language and Id beside talent Id.
6. The Language can be counted as a Talent, the Talent can have many languages or none.
7. The single Language is counted as one talent.
8. The website can have a Conference Participation part that holds Id, name , description, and date.
9. The user can enter his/her Volunteering information holding Id, Organization participated in, brief Explanation, and date.
10. The website user will have a Hobby section that will hold an Id and two different explanation parts like explanation1 and explanation2.
11. The website will count the Language, Conference Participation, and Volunteering as a Hobby at the same time, besides that Hobby can depend on other statements which can be added to the section.
12. Each user must relate to the About section which holds the user's Id, name, surname, address, telephone number, profession, photo, and description.
13. A Contact section is presented and holds Id, Name surname, Mail, Message subject, Message, date, and the user's Id to be sent in.
14. Each user might have many contact or none
15. While each contact must relate to only one user.
16. Education section is included in the system and holds Id, Title, two Subtitle sections, GPA, and date.
17. Each Education can be taken by at least one, or many users.
18. Each User can have many Education or none.
19. Each education must be taken by at least one user.

Entity Relationship model (Crow's Foot, Inheritance)



Relational model (textual representation)

- Experience (ID:integer, Title:character varying(40) Subtitle: character varying(40), Date:date, Number: integer)
- Talent (ID: integer, talent: character varying(40), talent_count: integer, experienceId: integer)
- Hobby(ID:integer, Explanation1: character varying(100), Explanation2: character varying(100),Language, Conference_Participation, Volunteering)
- Language(ID:integer, Language: character varying(30), talentId: integer)
- Conference_Participation(ID:integer, Conference_Name:character varying(30), Conference_Description:character varying(100), Conference_Date:date)
- Volunteering(ID:integer , Organization:character varying(30), Experience_Explanation:character varying(100), Date:date)
- About(ID:integer, Name:character varying(30), Surname:character varying(30), Address:character varying(40), TelephoneNo:integer, Profession:character varying(30), Photo: character varying, Description:character varying(100))
- Contact(ID:integer, NameSurname:character varying(40), Mail: character varying ,Subject:character varying, Message: character varying(100), Date:date, aboutId:integer)

- Education(ID:integer, Title:character varying, Subtitle1: character varying, Subtitle2:character varying, GPA:integer, Date:date)

SQL statements to create the database with the data in it

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 14.5
-- Dumped by pg_dump version 14.5

-- Started on 2022-12-25 21:03:50

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

DROP DATABASE "Hw_coredb";
--
-- TOC entry 3389 (class 1262 OID 24947)
-- Name: Hw_coredb; Type: DATABASE; Schema: -; Owner: postgres
--

CREATE DATABASE "Hw_coredb" WITH TEMPLATE = template0 ENCODING = 'UTF8' LOCALE =
'English_United States.1254';

ALTER DATABASE "Hw_coredb" OWNER TO postgres;

\connect "Hw_coredb"

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
```

```
SET client_min_messages = warning;
SET row_security = off;
```

```
--
-- TOC entry 221 (class 1255 OID 25288)
-- Name: hobby_update_trigger_fnc(); Type: FUNCTION; Schema: public; Owner: postgres
--
```

```
CREATE FUNCTION public.hobby_update_trigger_fnc() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    INSERT INTO "Hobby" ("ID", "Explanation1", "Explanation2")
        VALUES(NEW."id", NEW."exp1", NEW."exp2");
RETURN NEW;
END;
$$;
```

```
ALTER FUNCTION public.hobby_update_trigger_fnc() OWNER TO postgres;
```

```
--
-- TOC entry 224 (class 1255 OID 25296)
-- Name: increasegpa(); Type: FUNCTION; Schema: public; Owner: postgres
--
```

```
CREATE FUNCTION public.increasegpa() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
update "Education" set GPA=GPA+1;
return new;
end;
$$;
```

```
ALTER FUNCTION public.increasegpa() OWNER TO postgres;
```

```
--
-- TOC entry 223 (class 1255 OID 25293)
-- Name: increasenum(); Type: FUNCTION; Schema: public; Owner: postgres
--
```

```
CREATE FUNCTION public.increasenum() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
update "Experience" set Number=Number+1;
```

```
return new;  
end;  
$$;
```

```
ALTER FUNCTION public.increasenum() OWNER TO postgres;
```

```
--  
-- TOC entry 222 (class 1255 OID 25290)  
-- Name: increasetalent(); Type: FUNCTION; Schema: public; Owner: postgres  
--
```

```
CREATE FUNCTION public.increasetalent() RETURNS trigger  
    LANGUAGE plpgsql  
    AS $$  
BEGIN  
update "Talent" set talent_count=talent_count+1;  
return new;  
end;  
$$;
```

```
ALTER FUNCTION public.increasetalent() OWNER TO postgres;
```

```
SET default_tablespace = '';
```

```
SET default_table_access_method = heap;
```

```
--  
-- TOC entry 209 (class 1259 OID 24948)  
-- Name: About; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public."About" (  
    "ID" integer NOT NULL,  
    "Name" character varying(30) NOT NULL,  
    "Surname" character varying(30) NOT NULL,  
    "Address" character varying(40),  
    "TelephoneNo" integer NOT NULL,  
    "Profession" character varying(30),  
    "Photo" character varying,  
    "Description" character varying(100)  
);
```

```
ALTER TABLE public."About" OWNER TO postgres;
```

```
--
```

```
-- TOC entry 213 (class 1259 OID 24996)
-- Name: Conference_Participation; Type: TABLE; Schema: public; Owner: postgres
--
```

```
CREATE TABLE public."Conference_Participation" (
  "ID" integer NOT NULL,
  "Conference_Name" character varying(30) NOT NULL,
  "Conference_Description " character varying(100),
  "Conference_Date" date
);
```

```
ALTER TABLE public."Conference_Participation" OWNER TO postgres;
```

```
--
-- TOC entry 212 (class 1259 OID 24984)
-- Name: Contact; Type: TABLE; Schema: public; Owner: postgres
--
```

```
CREATE TABLE public."Contact" (
  "ID" integer NOT NULL,
  "NameSurname" character varying(40) NOT NULL,
  "Mail" character varying,
  "Subject" character varying,
  "Message" character varying(100),
  "Date" date NOT NULL,
  "aboutId" integer NOT NULL
);
```

```
ALTER TABLE public."Contact" OWNER TO postgres;
```

```
--
-- TOC entry 210 (class 1259 OID 24960)
-- Name: Education; Type: TABLE; Schema: public; Owner: postgres
--
```

```
CREATE TABLE public."Education" (
  "ID" integer NOT NULL,
  "Title" character varying(30) NOT NULL,
  "Subtitle1" character varying(30) NOT NULL,
  "Subtitle2" character varying(30) NOT NULL,
  "GPA" integer NOT NULL,
  "Date" date
);
```

```
ALTER TABLE public."Education" OWNER TO postgres;
```

```
--  
-- TOC entry 220 (class 1259 OID 25234)  
-- Name: Education_Life; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public."Education_Life" (  
    "Diploma_Number" character varying NOT NULL,  
    "Diploma_Name" character varying NOT NULL,  
    "Education_Summary " character varying,  
    "aboutId" integer NOT NULL,  
    "educationId" integer NOT NULL  
);
```

```
ALTER TABLE public."Education_Life" OWNER TO postgres;
```

```
--  
-- TOC entry 217 (class 1259 OID 25101)  
-- Name: experience_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres  
--
```

```
CREATE SEQUENCE public.experience_id_seq  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```
ALTER TABLE public.experience_id_seq OWNER TO postgres;
```

```
--  
-- TOC entry 218 (class 1259 OID 25113)  
-- Name: Experience; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public."Experience" (  
    "ID" integer DEFAULT nextval('public.experience_id_seq'::regclass) NOT NULL,  
    "Title" character varying(40),  
    "Subtitle " character varying(40),  
    "Date" date NOT NULL,  
    "Number" integer DEFAULT 1 NOT NULL  
);
```

```
ALTER TABLE public."Experience" OWNER TO postgres;
```



```
--  
-- TOC entry 211 (class 1259 OID 24972)  
-- Name: Hobby; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public."Hobby" (  
    "ID" integer NOT NULL,  
    "Explanation1" character varying(100),  
    "Explanation2" character(100)  
);
```

```
ALTER TABLE public."Hobby" OWNER TO postgres;
```

```
--  
-- TOC entry 215 (class 1259 OID 25011)  
-- Name: Language; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public."Language" (  
    "ID" integer NOT NULL,  
    "Language " character varying(30),  
    "talentId" integer NOT NULL  
);
```

```
ALTER TABLE public."Language" OWNER TO postgres;
```

```
--  
-- TOC entry 219 (class 1259 OID 25202)  
-- Name: Talent; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public."Talent" (  
    "ID" integer NOT NULL,  
    "Talent" character varying(40),  
    talent_count integer DEFAULT 1 NOT NULL,  
    "experienceId" integer  
);
```

```
ALTER TABLE public."Talent" OWNER TO postgres;
```

```
--  
-- TOC entry 214 (class 1259 OID 25006)  
-- Name: Volunteering; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public."Volunteering" (  
    "ID" integer NOT NULL,  
    "Organization" character varying(30) NOT NULL,  
    "Experience_Explanation " character varying(100),  
    "Date" date  
);
```

```
ALTER TABLE public."Volunteering" OWNER TO postgres;
```

```
--  
-- TOC entry 216 (class 1259 OID 25078)  
-- Name: __EFMigrationsHistory; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public."__EFMigrationsHistory" (  
    "MigrationId" character varying(150) NOT NULL,  
    "ProductVersion" character varying(32) NOT NULL  
);
```

```
ALTER TABLE public."__EFMigrationsHistory" OWNER TO postgres;
```

```
--  
-- TOC entry 3212 (class 2606 OID 24954)  
-- Name: About About_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public."About"  
    ADD CONSTRAINT "About_pkey" PRIMARY KEY ("ID");
```

```
--  
-- TOC entry 3220 (class 2606 OID 25000)  
-- Name: Conference_Participation Conference_Participation_pkey; Type: CONSTRAINT; Schema: public;  
Owner: postgres  
--
```

```
ALTER TABLE ONLY public."Conference_Participation"  
    ADD CONSTRAINT "Conference_Participation_pkey" PRIMARY KEY ("ID");
```

```
--  
-- TOC entry 3218 (class 2606 OID 24990)  
-- Name: Contact Contact_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public."Contact"
```

```

ADD CONSTRAINT "Contact_pkey" PRIMARY KEY ("ID");

--
-- TOC entry 3214 (class 2606 OID 24964)
-- Name: Education Education_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Education"
    ADD CONSTRAINT "Education_pkey" PRIMARY KEY ("ID");

--
-- TOC entry 3232 (class 2606 OID 25240)
-- Name: Education_Life Education_Life_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Education_Life"
    ADD CONSTRAINT "Education_Life_pkey" PRIMARY KEY ("Diploma_Number");

--
-- TOC entry 3228 (class 2606 OID 25118)
-- Name: Experience Experience_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Experience"
    ADD CONSTRAINT "Experience_pkey" PRIMARY KEY ("ID");

--
-- TOC entry 3216 (class 2606 OID 24976)
-- Name: Hobby Hobby_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Hobby"
    ADD CONSTRAINT "Hobby_pkey" PRIMARY KEY ("ID");

--
-- TOC entry 3224 (class 2606 OID 25015)
-- Name: Language Language_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Language"
    ADD CONSTRAINT "Language_pkey" PRIMARY KEY ("ID");

```

```
--
-- TOC entry 3226 (class 2606 OID 25082)
-- Name: _EFMigrationsHistory PK__EFMigrationsHistory; Type: CONSTRAINT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE ONLY public."__EFMigrationsHistory"
    ADD CONSTRAINT "PK__EFMigrationsHistory" PRIMARY KEY ("MigrationId");
```

```
--
-- TOC entry 3230 (class 2606 OID 25206)
-- Name: Talent Talent_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public."Talent"
    ADD CONSTRAINT "Talent_pkey" PRIMARY KEY ("ID");
```

```
--
-- TOC entry 3222 (class 2606 OID 25010)
-- Name: Volunteering Volunteering_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public."Volunteering"
    ADD CONSTRAINT "Volunteering_pkey" PRIMARY KEY ("ID");
```

```
--
-- TOC entry 3244 (class 2620 OID 25294)
-- Name: Talent trig1; Type: TRIGGER; Schema: public; Owner: postgres
--
```

```
CREATE TRIGGER trig1 AFTER INSERT ON public."Talent" FOR EACH ROW EXECUTE FUNCTION
public.increasenum();
```

```
--
-- TOC entry 3242 (class 2620 OID 25291)
-- Name: Language trig3; Type: TRIGGER; Schema: public; Owner: postgres
--
```

```
CREATE TRIGGER trig3 AFTER INSERT ON public."Language" FOR EACH ROW EXECUTE FUNCTION
public.increasetalent();
```

```
--
-- TOC entry 3241 (class 2620 OID 25297)
```

-- Name: About triggergpa; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER triggergpa AFTER INSERT ON public."About" FOR EACH ROW EXECUTE FUNCTION
public.increasegpa();

--

-- TOC entry 3243 (class 2620 OID 25289)

-- Name: Experience verify_for_update; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER verify_for_update BEFORE UPDATE ON public."Experience" FOR EACH ROW EXECUTE
FUNCTION public.hobby_update_trigger_fnc();

--

-- TOC entry 3239 (class 2606 OID 25257)

-- Name: Education_Life aboutid; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Education_Life"
ADD CONSTRAINT aboutid FOREIGN KEY ("aboutid") REFERENCES public."About"("ID");

--

-- TOC entry 3233 (class 2606 OID 25267)

-- Name: Contact aboutid; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Contact"
ADD CONSTRAINT aboutid FOREIGN KEY ("aboutid") REFERENCES public."About"("ID");

--

-- TOC entry 3234 (class 2606 OID 25277)

-- Name: Conference_Participation conferencefk; Type: FK CONSTRAINT; Schema: public; Owner:
postgres

--

ALTER TABLE ONLY public."Conference_Participation"
ADD CONSTRAINT conferencefk FOREIGN KEY ("ID") REFERENCES public."Hobby"("ID");

--

-- TOC entry 3235 (class 2606 OID 25282)

-- Name: Volunteering conferencefk; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

```

ALTER TABLE ONLY public."Volunteering"
  ADD CONSTRAINT conferencefk FOREIGN KEY ("ID") REFERENCES public."Hobby"("ID");

--
-- TOC entry 3240 (class 2606 OID 25262)
-- Name: Education_Life educationid; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Education_Life"
  ADD CONSTRAINT educationid FOREIGN KEY ("educationId") REFERENCES public."Education"("ID");

--
-- TOC entry 3238 (class 2606 OID 25242)
-- Name: Talent experienceid; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Talent"
  ADD CONSTRAINT experienceid FOREIGN KEY ("experienceId") REFERENCES public."Experience"("ID");

--
-- TOC entry 3237 (class 2606 OID 25272)
-- Name: Language languagefk; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Language"
  ADD CONSTRAINT languagefk FOREIGN KEY ("ID") REFERENCES public."Hobby"("ID");

--
-- TOC entry 3236 (class 2606 OID 25247)
-- Name: Language talentid; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Language"
  ADD CONSTRAINT talentid FOREIGN KEY ("talentId") REFERENCES public."Talent"("ID");

-- Completed on 2022-12-25 21:03:51

--
-- PostgreSQL database dump complete
--

```

```
create or replace function increaseGpa()  
returns TRIGGER  
as  
$$  
BEGIN  
update "Education" set GPA=GPA+1;  
return new;  
end;  
$$  
language plpgsql;  
  
create trigger triggerGpa  
after INSERT  
on "About"  
for each row  
execute procedure increaseGpa();
```

```
create or replace function increaseNum()
returns TRIGGER
as
$$
BEGIN
update "Experience" set Number=Number+1;
return new;
end;
$$
language plpgsql;
```

```
create trigger trig1
after INSERT
on "Talent"
for each row
execute procedure increaseNum();
```

```
CREATE OR REPLACE FUNCTION hobby_update_trigger_fnc()
RETURNS trigger AS
$$
BEGIN
    INSERT INTO "Hobby" ("ID","Explanation1", "Explanation2")
        VALUES (NEW."id",NEW."exp1",NEW."exp2");
RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';

CREATE TRIGGER verify_for_update
BEFORE UPDATE
ON "Experience"
FOR EACH ROW
EXECUTE PROCEDURE hobby_update_trigger_fnc();
```

```

create or replace function increaseTalent()
returns TRIGGER
as
$$
BEGIN
update "Talent" set talent_count=talent_count+1;
return new;
end;
$$
language plpgsql

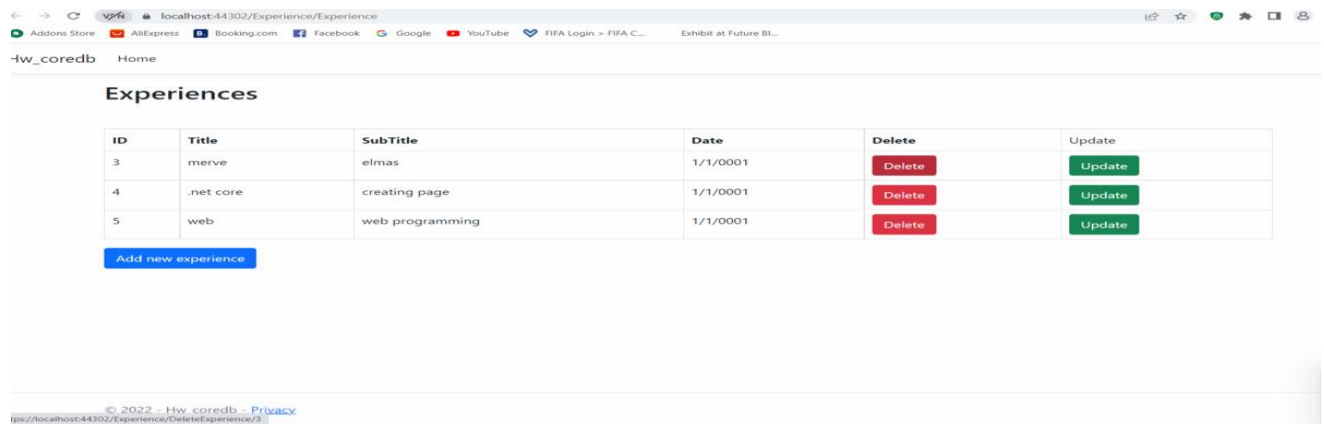
create trigger trig3
after INSERT
on "Language"
for each row
execute procedure increaseTalent();

```

Screenshots of (Search, Insert, Delete, Update)

1-DELETE: When the user select delete button you can see that it will delete the element like the example below.

- *Before deleting*

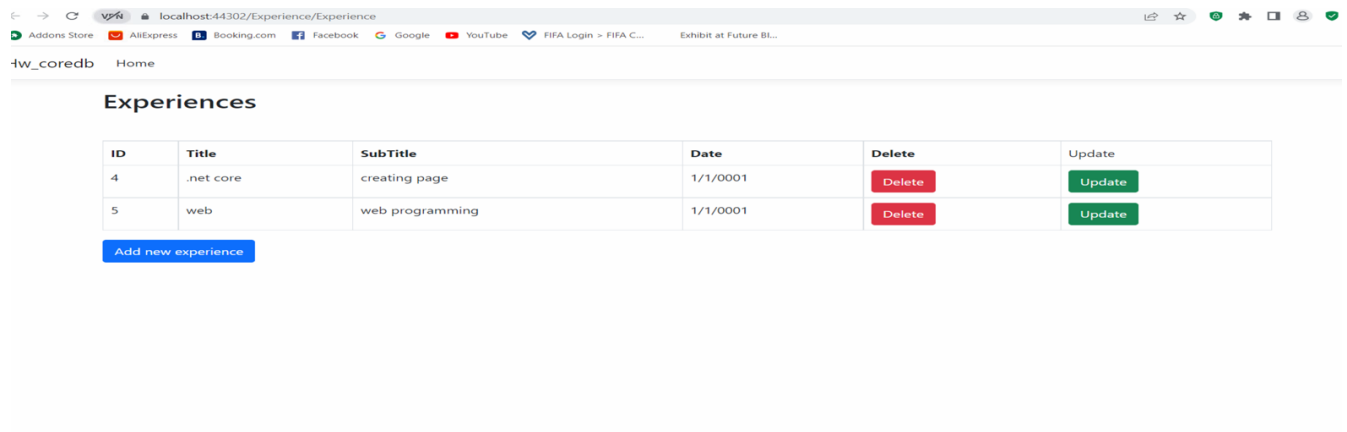


The screenshot shows a web browser window with the URL `localhost:44302/Experience/Experience`. The page displays a table titled "Experiences" with the following data:

ID	Title	SubTitle	Date	Delete	Update
3	nerve	elmas	1/1/0001	<button>Delete</button>	<button>Update</button>
4	.net core	creating page	1/1/0001	<button>Delete</button>	<button>Update</button>
5	web	web programming	1/1/0001	<button>Delete</button>	<button>Update</button>

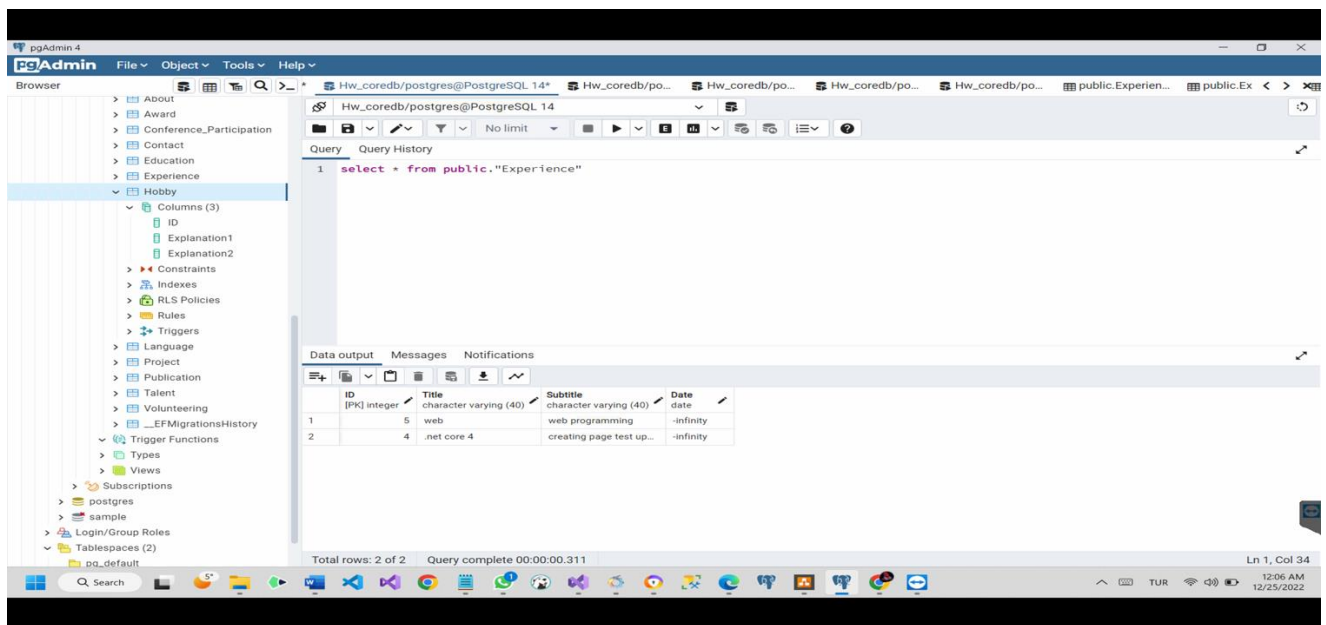
Below the table is a blue button labeled "Add new experience". At the bottom of the page, there is a footer with the text: "© 2022 - Hw_coredb - Privacy" and a link to "https://localhost:44302/Experience/Delete/Experience/3".

- *After delete operation*



After deleting the operation from the web app you can see that there is no data on the database.

- Database section after deleting



2-UPDATE

- Updating data from the web app

← → ↻ localhost:44302/Experience/UpdateExperience/4

[Addons Store](#) [AliExpress](#) [Booking.com](#) [Facebook](#) [Google](#) [YouTube](#) [FIFA Login > FIFA C...](#) [Exhibit at Future BL...](#)

HW_coredb Home

Updating

Title

SubTitle

Date

[Update](#)

© 2022 - HW_coredb - [Privacy](#)

- You can see that there is a differences on the table

← → ↻ localhost:44302/Experience/Experience

[Addons Store](#) [AliExpress](#) [Booking.com](#) [Facebook](#) [Google](#) [YouTube](#) [FIFA Login > FIFA C...](#) [Exhibit at Future BL...](#)

HW_coredb Home

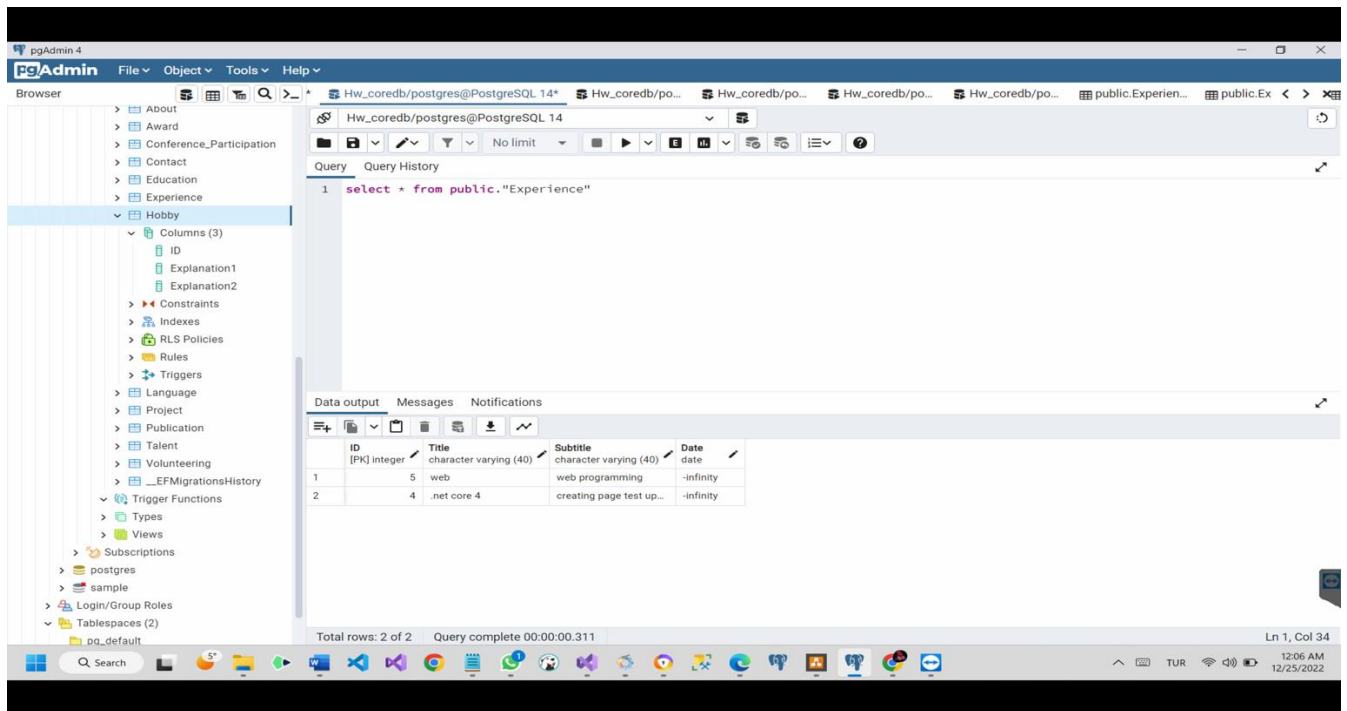
Experiences

ID	Title	SubTitle	Date	Delete	Update
5	web	web programming	1/1/0001	Delete	Update
4	.net core 4	creating page test update	1/1/0001	Delete	Update

[Add new experience](#)

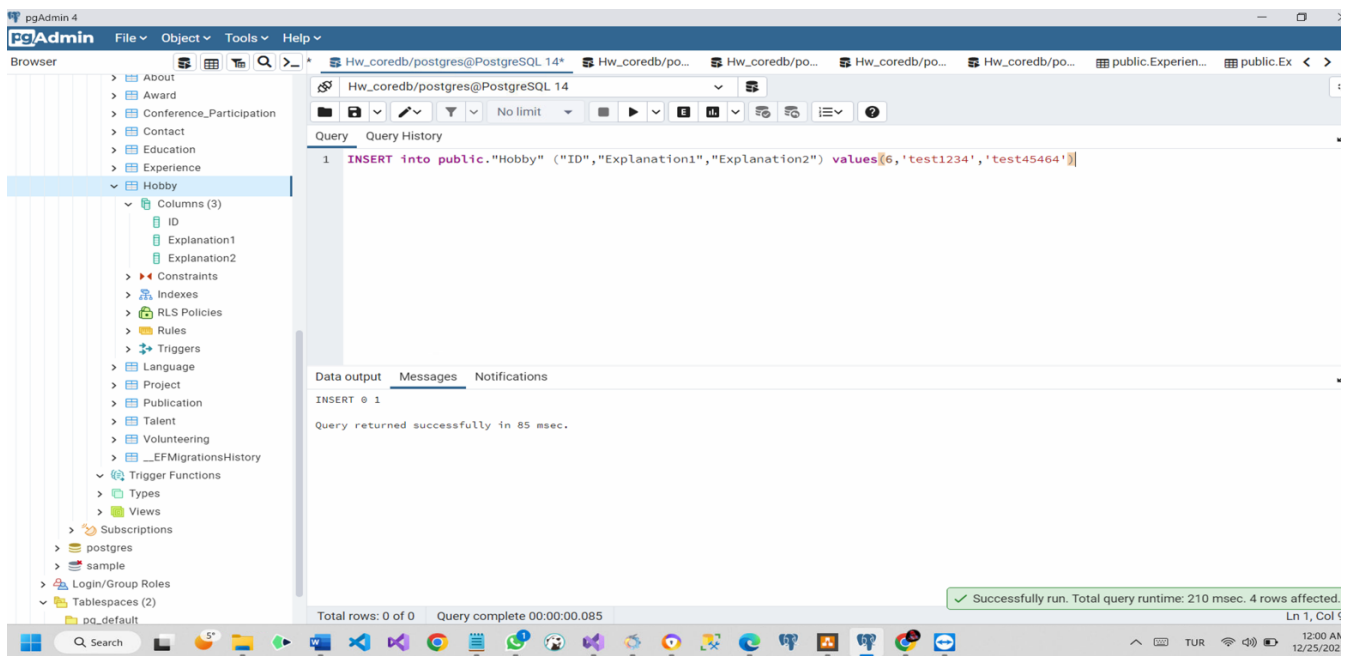
© 2022 - HW_coredb - [Privacy](#)

- Let's look to the database section about any differences.

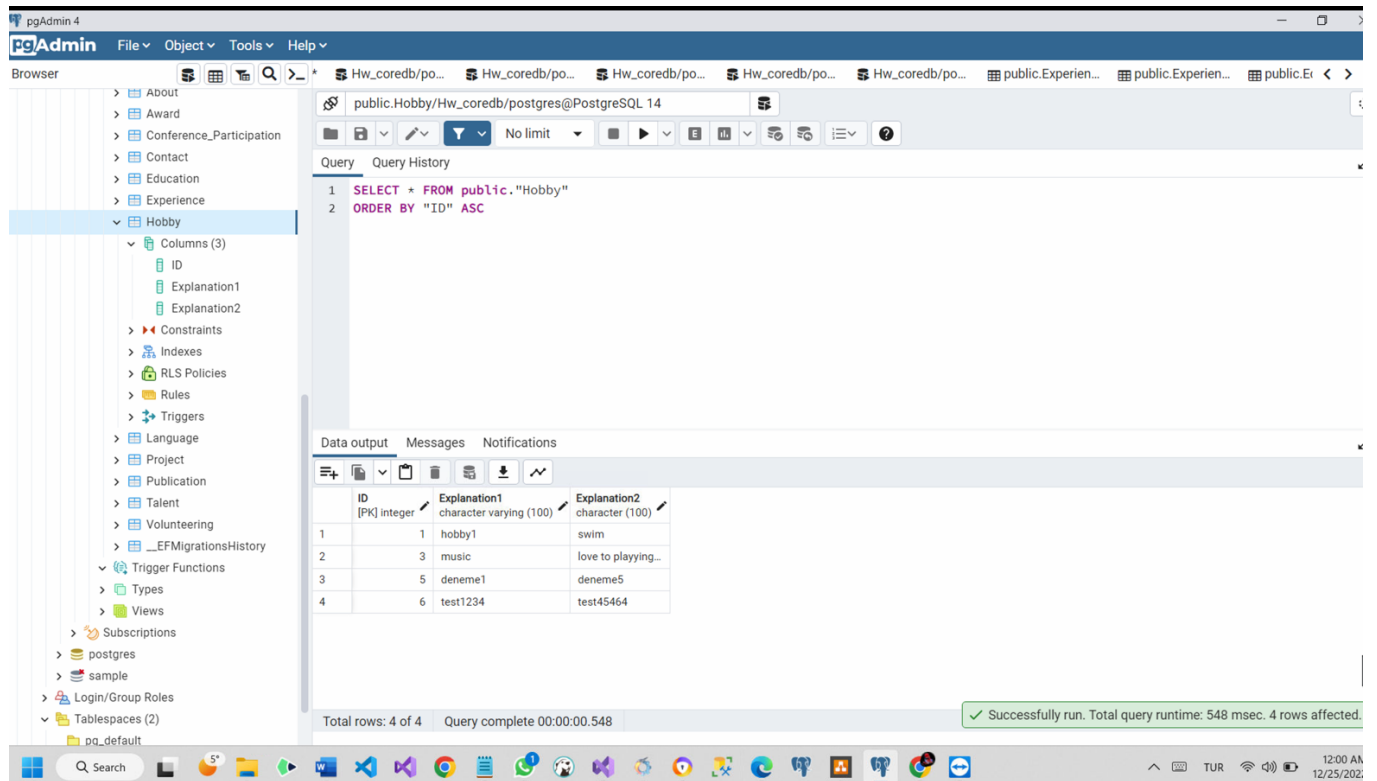


3-INSERT DATA

- When the user insert into to the database



- User can see the inserting values on the database also we can check it from the web app.



- web hobby screenshot

