Introduzione alla programmazione WEB

AA 2015 – 2016

Docenti: Gino Perna e Stefano Chirico

Progetto finale

"Sistema di recensione e classificazione di ristoranti"

Versione documento: 1.6~~3~~ – 2016-05-~~15~~27

SCOPO

Si vuole progettare un sistema di recensione e classificazione di ristoranti da esporre via WEB e fruibile per dispositivi sia desktop che mobile in modalità responsive. L'applicazione deve essere sviluppata con tecnolgia Servlet/JSP/JSTL ed ove fosse conveniente a vostra scelta WEBservices,CSS, javascript

Il sistema prevede:

- Un motore di ricerca che permetta di visualizzare un elenco di ristoranti in base a varie tipologia di scelta tra cui dovranno esserci:
  - Per zona geografica (elenco di ristoranti per stato, città, provincia);
  - Per tipologia di cucina (elenco di ristoranti per cucina etnica, italiana, ecc…);
  - Per nome (elenco di ristoranti che abbiano nome simile a quello ricercato).
  - Per indirizzo [5+] inteso come via e numero civico
- Un sistema di classificazione dei ristoranti che dipenda dal giudizio degli utenti del portale dal numero di recensioni totali di ogni ristorante e~~-~~, dalla loro reputazione [quest'ultima 5+].
- Un sistema di classificazione degli utenti (reputazione) che dipenda dall'utilità delle recensioni lasciate. [5+]
- La possibilità, per un utente registrato, di inserire uno o più nuovi ~~nuovo~~ ristorante [5+].
- La possibilità, per un utente registrato, di reclamare un ristorante esistente (è possibile avere più di un ristorante contemporaneamente).
- La possibilità, per gli utenti registrati, di lasciare una o più recensioni ad uno o più ristoranti.
- La possibilità, per i proprietari di ristoranti, di rispondere alle recensioni che i clienti lasciano per i propri ristoranti. Tale risposta verrà visualizzata sul sito solo dopo controllo e autorizzazione da parte di un amministratore del portale.[5+]
- La possibilità, per un utente registrato, di inserire fotografie legate ad uno o più ristoranti.
- La possibilità, per i proprietari di ristoranti, di segnalare fotografie non consone. Tale segnalazione verrà valutata da parte di un amministratore del portale.~~[5+]~~ che potra' eliminare le foto
- Un sistema di georeferenziazione che permette di visualizzare, su una mappa, i ristoranti vicini a quello selezionato durante la ricerca.[5+]
- L'internazionalizzazione del sito in base alle informazioni ricevute dal browser e/o scelta diretta dell'utente [7] (inteso come localizzazione dei menu e delle informazioni/headers tabelle e bottoni; non devono essere localizzate ovviamente le recensioni che rimangono nella lingua originale). L'internazionalizzazione è intesa che segua le regole spiegate a lezione o superiori (essendo attivi formalmente uno standard de-facto presentato ed uno standard dell'IANA che codifica le nazioni a due o più lettere)

RISTORANTE

I ristoranti saranno classificati almeno da:

- Nome;
- Descrizione;
- Fotografie pubblicate dal proprietario del ristorante;
- Link al sito ufficiale;
- Tipologie di cucina;
- Coordinate geografiche (città, indirizzo, ecc…);
- Orari d'apertura;
- Fascia di prezzo (economici, medi, costosi);
- Valutazione totale (da 0 a 5) (il voto e' inteso come voto globale dato al ristorante).

UTENTI

Gli utenti saranno classificati almeno da:

- Nome;
- Cognome;
- Email;
- Password.
- AVATAR [7+]

Il portale dovrà prevedere quattro tipologie di utenti:

1. Utente anonimo;
2. Utente registrato;
3. Utente ristoratore;
4. Utente amministratore.

UTENTE ANONIMO

Tale utente è quello che accede al sito senza fare login all'interno dello stesso. Tale utente può eseguire le ricerche (come già descritte in SCOPO). L'utente anonimo potrà registrarsi al portale via FORM di registrazione.

UTENTE REGISTRATO

Tale utente è quello che ha fatto login al sito. L'utente registrato è quello che esegue la maggior parte delle iterazioni con il portale web. Oltre a poter svolgere le attività dell'utente anonimo potrà: dare uno o più voti ad un ristorante (in giorni diversi); lasciare recensioni; caricare foto; inserire ristoranti. [~~modificare~~Modificare/immettere il proprio avatar che deve essere visualizzato nelle recensioni 7+]

UTENTE RISTORATORE

Un utente registrato potrà reclamare un ristorante esistente (anche inserito da altri utenti registrati). Nel caso tale ristorante gli venga assegnato da un utente amministratore, allora tale utente diventerà automaticamente UTENTE RISTORATORE e potrà: rispondere alle recensioni degli utenti registrati; segnalare foto non consone; modificare i dati relativi al ristorante; Inoltre, potrà eseguire tutte le attività di un utente registrato.

UTENTE AMMINISTRATORE

Tale utente è il gestore del sito web e si occupa di gestire le iterazioni tra utenti registrati e utenti ristoratori. Tale utente verrà notificato di nuove risposte da parte di un ristoratore per cui dovrà autorizzarne la visibilità sul portale; verrà notificato di segnalazioni di fotografie non consone per cui dovrà

autorizzarne la rimozione dal portale; verrà notificato della richiesta di assegnazione di un ristorante per cui dovrà valutarne la validità e, quindi, associare l'utente al ristoratore.

USE CASE (lo use case e' riferito all'intera applicazione, in funzione del numero di persone componenti il gruppo dovete verificare che lo usecase sia congruente con qunto richiesto)

Menu principale

Ogni pagina del portale dovrà avere lo stesso menu principale personalizzato per tipologia di utente.

L'***utente anonimo*** avrà accesso al seguente menu principale:

- Pulsante "Registrati": che lo farà accedere alla form di registrazione;
- Pulsante "Accedi": che lo farà accedere alla form di login;
- Pulsante "Lingua": che gli permetterà di modificare l'internazionalizzazione del portale. (dovranno essere presenti almeno due lingue ed un locale di default. Attenzione a fare una traduzione di tutte le keywords (come spiegato a lezione con GoogleTranslator) in modo che sia evidente all'utilizzatore la lingua scelta e non di solo alcune)

L'***utente registrato*** avrà accesso al seguente menu principale:

- Pulsante "Nome e Cognome";
    - Profilo: che permetterà di modificare i dati di profilo dell'utente;
    - Aggiungi ristorante: che permette di aggiungere un nuovo ristorante;
    - Esci: che permetterà di fare logout dell'utente.
- Pulsante "Lingua": che gli permetterà di modificare l'internazionalizzazione del portale.

L'***utente ristoratore*** avrà accesso al seguente menu principale:

- Pulsante "Notifiche": visualizzerà l'elenco delle ultime notifiche ricevute più un pulsante per accedere a tutte le notifiche;
- Pulsante "Nome e Cognome";
    - Profilo: che permetterà di modificare i dati di profilo dell'utente;
    - Il vostro ristorante: che permetterà di accedere alla pagina di gestione del proprio ristorante;
    - Esci: che permetterà di fare logout dell'utente.
- Pulsante "Lingua": che gli permetterà di modificare l'internazionalizzazione del portale.

L'utente amministratore avrà accesso al seguente menu principale:

- Pulsante "Notifiche": visualizzerà l'elenco delle ultime notifiche ricevute più un pulsante per accedere a tutte le notifiche;
- Pulsante "Nome e Cognome";
    - Profilo: che permetterà di modificare i dati di profilo dell'utente;
    - Esci: che permetterà di fare logout dell'utente.
- Pulsante "Lingua": che gli permetterà di modificare l'internazionalizzazione del portale.

Landing Page

Ogni tipologia di utente avrà accesso alla landing page del portale web che sarà composto da:

- Form di ricerca:
    - Text field con auto-completamento: che permetterà di inserire le parole da utilizzare per la ricerca dei ristoranti;
    - Pulsante "Cerca": che avvierà la ricerca dei ristoranti.
- Elenco di possibili scelte:

    - Elenco di opzioni per la ricerca di ristoranti per tipologia vicini a dove si trova l'utente;
    - Elenco di opzioni per la ricerca di ristoranti più richiesti vicini a dove si trova l'utente.

Form di registrazione

L'utente anonimo avrà accesso a tale form che dovrà permettere la registrazione dell'utente e dovrà prevedere:

- Email: che verrà utilizzato anche come username di login;
- Password: che dovrà avere dei meccanismi di scelta di una password solida;
- L'accettazione (anche non immediata, ma prima della prima connessione) alla normativa sulla privacy da accettare;
- Pulsante "Iscriviti": che invierà la richiesta d'iscrizione;
- Pulsante "Annulla": che rimanderà l'utente alla pagina da cui era arrivato.

L'iscrizione al portale dovrà essere validato attraverso il click di un link inviato per email.

Form di login

L'utente anonimo avrà accesso a tale form che dovrà permettere il login dell'utente e dovrà prevedere:

- Username: l'email con cui l'utente si è registrato;
- Password: la password con cui l'utente si è registrato;
- Modalita' reset della password se l'utente se l'è dimenticata;
- Pulsante "Accedi": che effettuerà il login dell'utente e ritornerà alla pagina di provenienza;
- Pulsante "Annulla": che rimanderà l'utente alla pagina da cui era arrivato.

Form di ricerca

Ogni tipologia di utente avrà accesso a tale form che dovrà permettere la ricerca di ristorante e dovrà prevedere:

- Text field (uno o piu' consigliati due stile trip advisor): verranno inserite le parole da ricercare;
- Pulsante "Cerca": avvierà la ricerca e rimanderà l'utente alla pagina dei risultati.

Il campo di ricerca avrà l'auto completamento che aiuterà l'utente nella scelta delle parole da ricercare.

La ricerca dovra' poter includere almeno: il nome ristorante, la via/citta' /regione.

Pagina dei risultati

Ogni tipologia di utente avrà accesso alla pagina dei risultati che dovrà prevedere:

- L'elenco dei ristoranti ordinati per classifica; per ogni ristorante dovrà essere visibile:
    - Nome del ristorante con link alla pagina del ristorante;
    - Una foto;
    - Il voto totale (da 1 a 5);
    - La posizione in classifica;
    - Il numero di recensioni ricevute;
    - Le tipologie di cucina;
    - Link alla mappa.
- Possibilità di modificare l'ordinamento della classifica:
    - Posizione in classifica;
    - Alfabetico;
    - Fascia di prezzo.
- Possibilità di filtrare l'elenco:
    - Tipologia di cucina;

- Fascia di prezzo.

Pagina della mappa

Dovrà visualizzare una mappa di Google centrata sul ristorante selezionato[per tutti] e [seguente per 5+] che presenti tutti i ristoranti presenti in zona.

Pagina del ristorante

Questa pagina dovrà avere almeno le seguenti informazioni:

- Nome;
- Valutazione totale;
- Posizione in classifica (per città);
- Tipologie di cucina;
- Fascia di prezzo;
- Orari d'apertura;
- indirizzo
- Almeno una fotografia;
- Elenco delle ultime recensioni;
- Un QR conentente le seguenti informazioni: Nome, indirizzo, orari di apertura
- Link per reclamare la proprietà del ristorante;

Pagina delle notifiche

L'utente ristoratore visualizzerà, in questa pagina, tutte le notifiche di avvenuta recensione o di caricamento di nuove fotografie.

Da ogni notifica potrà accedere, rispettivamente, a:

- La pagina di risposta alle recensioni;
- La pagina di richiesta di rimozione di fotografie.

L'utente amministratore visualizzerà, in questa pagina, tutte le notifiche di avvenuta risposta ad una recensione o di richiesta di rimozione di fotografie da parte di un ristoratore.

Per ogni notifica potrà:

- Validare o meno la richiesta;
- Accedere alla risposta per validarla o non validarla;
- Accedere alla fotografia per rimuoverla o non rimuoverla.

Precompilazione database

Ciascun progetto(gruppo) ~~dovra'~~dovrà presentare il proprio lavoro con un database ~~gia'~~già popolato per almeno:

- ~~Tre~~ Due diverse aree geografiche per almeno 5 ~~citta'~~città differenti (anche in differenti stati se volete)
- Per ciascuna ~~citta'~~città (vista come area estesa attorno alla ~~citta'~~città) almeno 10 ristoranti diversi.
- Per ciascuna citta' la somma delle recensioni (per il totale dei 10 ristoranti) dovra' essere di almeno 20. Dovranno inoltre essere presenti almeno 4 ristoranti per citta' con foto pubblicate da utenti.
- ~~Ciascun ristorante dovra' avere almeno quattro recensioni di cui almno una con foto pubblicate da un utente~~
- Almeno l'80% dei ristoranti presenti deve essere stato reclamato da un ristoratore ed avere una foto del ristorante

- 10 utenti minimo

Database

Per agevolare chi non avesse ancora completato il corso di DB viene di seguito indicato un possibile diagramma ER di un database che, ragionevolmente, sostiene l'applicazione.

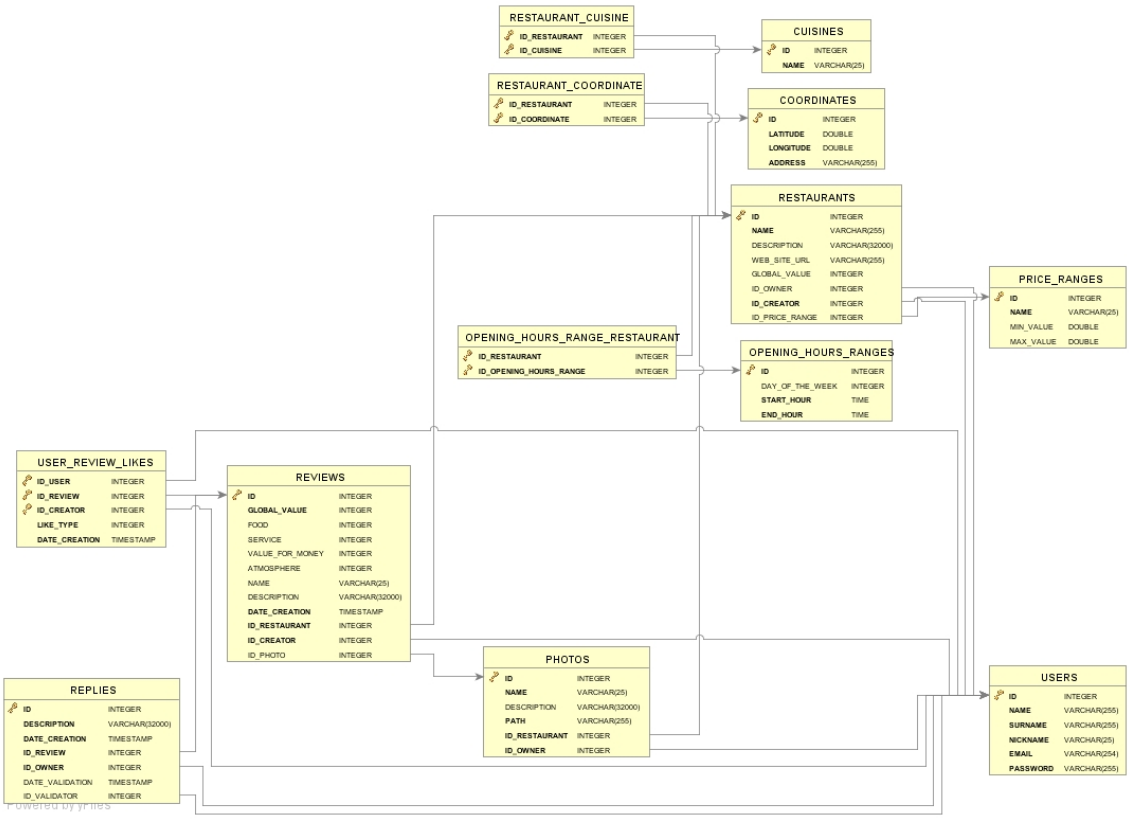

Figura 1: Diagramma ER database

Il DB copre le intere specifiche descritte precedentemente. Il contenuto delle tabelle è volutamente ridotto al minimo ed è indicativo. Anche la struttura del database stesso è del tutto indicativa e può essere modificata a piacimento secondo le vostre esigenze (ovviamente bisogna saper sostenere le scelte fatte).

Hints vari

- String similarity: https://github.com/tdebatty/java-string-similarity. Si tenga presente che per una ricerca efficiente (i.e. motore di ricerca vengono impiegati algoritmi che pre-memorizzano l'indice scelto e quindi ne fanno una sola comparazione al momento della query).
- Geocoding:
  https://developers.google.com/maps/documentation/geocoding/intro#GeocodingRequests e
  https://developer.mapquest.com/plan_purchase/steps/business_edition/business_edition_free/register + http://www.mapquestapi.com/geocoding/
  Ricordarsi che bisogna prememorizzare le coordinate geografiche o box per poi fare la ricerca su numeri.
- Riempite il database con dati plausibili, evitate nomi tipo 'ristorante1 ristorante 2' etc… cercate di rendere presentabile e credibile l'implementazione (in totale: 2*5* 20 recensioni, suggerimenti per recuperare recensioni/ ricerca: guida michelin, tripadvisor, yelp.com, timeout.com http://www.gnavi.co.jp/en/
-

Raccomandazioni

Il sito deve essere progettato in modalita’ responsive e dimostrato funzionante per risoluzioni pc/tablet/telefono attraverso il device o la modalita’ relativa sul browser

Il progetto dovra' essere mostrato con un database minimo credibile ed intelligentemente popolato per quanto richiesto e nel numero di voci richieste nel presente testo come minimo.

Documentazione

Dovra’ essere presentata una documentazione del progetto illustrante le caratteristiche e le scelte effettuate

Ricordarsi che per una buona valutazione:

- all'esame servono due browser per dimostrare le funzionalita'
- gli uri di tutti i files devono essere relativi e non assoluti sul file system
- La documentazione presentata deve essere efficace
- Attenzione a chi ~~usera'~~userà javascript: deve sapere sostenere le scelte implementative *tutti i componenti del gruppo, e non deve violare o rendere inutile il paradigma MVC
- Il database con cui implementare il progetto e' a libera scelta (postgres/mysql/derby/ etc...), ricordandosi che nella consegna ~~dovra'~~dovrà essere data una copia anche del database

Regole di presentazione e consegna progetto

TBD, indicativamente consegna progetto 2 giorni prima dell'esame, da un solo componente del gruppo con chiara indicazione dei partecipanti al gruppo. Consegna progetto elettronica tramite link a strumento di archiviazione cloud a vostra scelta

Dare un NIKNAME al progetto in modo che sia identificabile facilmente rispetto agli altri

SCHEMA DATABASE

Progetto finale query tabelle db.ddl

```
CREATE TABLE USERS (
        ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
        NAME VARCHAR(255) NOT NULL,
        SURNAME VARCHAR(255) NOT NULL,
        NICKNAME VARCHAR(25) NOT NULL,
        EMAIL VARCHAR(254) NOT NULL,
        PASSWORD VARCHAR(255) NOT NULL,
PRIMARY KEY (ID),
UNIQUE (NICKNAME),
UNIQUE (EMAIL)
);

CREATE TABLE PRICE_RANGES (
        ID INTEGER NOT NULL,
        NAME VARCHAR(25) NOT NULL,
        MIN_VALUE FLOAT,
        MAX_VALUE FLOAT,
PRIMARY KEY (ID),
UNIQUE (NAME),
CONSTRAINT PRICE_RANGE_CHECK CHECK ((MIN_VALUE IS NULL AND MAX_VALUE IS NOT
NULL) OR (MIN_VALUE IS

NOT NULL AND MAX_VALUE IS NULL) OR (MIN_VALUE < MAX_VALUE))
);

CREATE TABLE RESTAURANTS (
        ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
        NAME VARCHAR(255) NOT NULL,
        DESCRIPTION VARCHAR(32000),
        WEB_SITE_URL VARCHAR(255),
        GLOBAL_VALUE INTEGER CONSTRAINT GLOBAL_VALUE_CHECK CHECK (GLOBAL_VALUE
>= 0 AND

GLOBAL_VALUE <= 5),
```

```
        ID_OWNER INTEGER,
            ID_CREATOR INTEGER NOT NULL,
        ID_PRICE_RANGE INTEGER,
PRIMARY KEY (ID),
FOREIGN KEY (ID_OWNER) REFERENCES USERS (ID),
FOREIGN KEY (ID_CREATOR) REFERENCES USERS (ID),
FOREIGN KEY (ID_PRICE_RANGE) REFERENCES PRICE_RANGES (ID)
);

CREATE TABLE PHOTOS (
        ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
        NAME VARCHAR(25) NOT NULL,
        DESCRIPTION VARCHAR (32000),
        PATH VARCHAR(255) NOT NULL,
        ID_RESTAURANT INTEGER NOT NULL,
        ID_OWNER INTEGER NOT NULL,
PRIMARY KEY (ID),
FOREIGN KEY (ID_RESTAURANT) REFERENCES RESTAURANTS (ID),
FOREIGN KEY (ID_OWNER) REFERENCES USERS (ID)
);

CREATE TABLE COORDINATES (
        ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
        LATITUDE FLOAT NOT NULL,
        LONGITUDE FLOAT NOT NULL,
        ADDRESS VARCHAR(255) NOT NULL,
PRIMARY KEY (ID)
);

CREATE TABLE RESTAURANT_COORDINATE (
        ID_RESTAURANT INTEGER NOT NULL,
        ID_COORDINATE INTEGER NOT NULL,
PRIMARY KEY (ID_RESTAURANT, ID_COORDINATE),
FOREIGN KEY (ID_RESTAURANT) REFERENCES RESTAURANTS (ID),
FOREIGN KEY (ID_COORDINATE) REFERENCES COORDINATES (ID)
```

```
);

CREATE TABLE OPENING_HOURS_RANGES (
        ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
        DAY_OF_THE_WEEK INTEGER CONSTRAINT DAY_OF_THE_WEEK_CHECK CHECK
(DAY_OF_THE_WEEK >= 1 AND

DAY_OF_THE_WEEK <= 7),
        START_HOUR TIME NOT NULL,
        END_HOUR TIME NOT NULL,
PRIMARY KEY (ID),
CONSTRAINT RANGE_CHECK CHECK (START_HOUR < END_HOUR)
);

CREATE TABLE OPENING_HOURS_RANGE_RESTAURANT (
        ID_RESTAURANT INTEGER NOT NULL,
        ID_OPENING_HOURS_RANGE INTEGER NOT NULL,
PRIMARY KEY (ID_RESTAURANT, ID_OPENING_HOURS_RANGE),
FOREIGN KEY (ID_RESTAURANT) REFERENCES RESTAURANTS (ID),
FOREIGN KEY (ID_OPENING_HOURS_RANGE) REFERENCES OPENING_HOURS_RANGES (ID)
);

CREATE TABLE CUISINES (
        ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
        NAME VARCHAR(25) NOT NULL,
PRIMARY KEY (ID)
);

CREATE TABLE RESTAURANT_CUISINE (
        ID_RESTAURANT INTEGER NOT NULL,
        ID_CUISINE INTEGER NOT NULL,
PRIMARY KEY (ID_RESTAURANT, ID_CUISINE),
FOREIGN KEY (ID_RESTAURANT) REFERENCES RESTAURANTS (ID),
FOREIGN KEY (ID_CUISINE) REFERENCES CUISINES (ID)
);
```

```
CREATE TABLE REVIEWS (

        ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,

        GLOBAL_VALUE INTEGER NOT NULL CONSTRAINT REVIEW_GLOBAL_VALUE_CHECK CHECK
(GLOBAL_VALUE >= 1
AND GLOBAL_VALUE <= 5),

        FOOD INTEGER CONSTRAINT REVIEW_FOOD_CHECK CHECK (FOOD IS NULL OR (FOOD
>= 1 AND FOOD <=
5)),

        SERVICE INTEGER CONSTRAINT REVIEW_SERVICE_CHECK CHECK (SERVICE IS NULL
OR (SERVICE >= 1 AND
SERVICE <= 5)),

        VALUE_FOR_MONEY INTEGER CONSTRAINT REVIEW_VALUE_FOR_MONEY_CHECK CHECK
(VALUE_FOR_MONEY IS
NULL OR (VALUE_FOR_MONEY >= 1 AND VALUE_FOR_MONEY <= 5)),

        ATMOSPHERE INTEGER CONSTRAINT REVIEW_ATMOSPHERE_CHECK CHECK (ATMOSPHERE
IS NULL OR
(ATMOSPHERE >= 1 AND ATMOSPHERE <= 5)),

        NAME VARCHAR(25),

        DESCRIPTION VARCHAR(32000),

        DATE_CREATION TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,

        ID_RESTAURANT INTEGER NOT NULL,

        ID_CREATOR INTEGER NOT NULL,

        ID_PHOTO INTEGER,

PRIMARY KEY (ID),

FOREIGN KEY (ID_RESTAURANT) REFERENCES RESTAURANTS (ID),

FOREIGN KEY (ID_CREATOR) REFERENCES USERS (ID),

FOREIGN KEY (ID_PHOTO) REFERENCES PHOTOS (ID)

);


CREATE TABLE REPLIES (

        ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,

        DESCRIPTION VARCHAR(32000) NOT NULL,

        DATE_CREATION TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,

        ID_REVIEW INTEGER NOT NULL,

        ID_OWNER INTEGER NOT NULL,

        DATE_VALIDATION TIMESTAMP,

        ID_VALIDATOR INTEGER,

PRIMARY KEY (ID),

FOREIGN KEY (ID_REVIEW) REFERENCES REVIEWS (ID),
```

```
FOREIGN KEY (ID_OWNER) REFERENCES USERS (ID),
FOREIGN KEY (ID_VALIDATOR) REFERENCES USERS (ID)
);

CREATE TABLE USER_REVIEW_LIKES (
        ID_USER INTEGER NOT NULL,
        ID_REVIEW INTEGER NOT NULL,
        ID_CREATOR INTEGER NOT NULL,
        LIKE_TYPE INTEGER NOT NULL CONSTRAINT LIKE_TYPE_CHECK CHECK (LIKE_TYPE
>= 0 AND LIKE_TYPE

<= 1),
        DATE_CREATION TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (ID_USER, ID_REVIEW, ID_CREATOR),
FOREIGN KEY (ID_USER) REFERENCES USERS (ID),
FOREIGN KEY (ID_REVIEW) REFERENCES REVIEWS (ID),
FOREIGN KEY (ID_CREATOR) REFERENCES USERS (ID)
);
```