

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής



Εργασία μαθήματος «Βάσεις δεδομένων»

4^ο εξάμηνο

Στοιχεία φοιτητών	
Ονοματεπώνυμο	Αριθμός μητρώου
Παναγιώτης Μουρλάς	Π19108
Ευστράτιος Καρκάνης	Π19064
Χρήστος Σεραφείμ Νικητάκης	Π19121

Περιεχόμενα

1 Σχεδίαση της βάσης δεδομένων	3
1.1 Γενικές λεπτομέρειες για την βάση δεδομένων	3
1.2 Περιγραφή των σχέσεων της βάσης δεδομένων (ερώτημα 1α)	3
1.3 Το διάγραμμα ERD της βάσης δεδομένων (ερώτημα 1α)	15
1.4 Κανονικοποίηση της βάσης δεδομένων (ερώτημα 1b)	15
2 Εντολές SELECT (ερώτημα 2)	20
3 Υλοποίηση triggers και cursors (ερώτημα 3)	29
3.1 Κατασκευή trigger (ερώτημα 3a)	29
3.2 Κατασκευή cursor (ερώτημα 3b)	31
4 Σύνδεση βάσης δεδομένων με API (ερώτημα 4)	32

1 Σχεδίαση της βάσης δεδομένων

1.1 Γενικές λεπτομέρειες για την βάση δεδομένων

Για το συγκεκριμένο project, καλούμαστε να σχεδιάσουμε την βάση δεδομένων μίας εταιρίας ασφάλισης. Πιο συγκεκριμένα, η πληροφορία που διατηρείται σε αυτή την σχεσιακή βάση, περιλαμβάνει τα ασφαλιστικά συμβόλαια, τα οχήματα που σχετίζονται με αυτά τα συμβόλαια, τους οδηγούς και τους πελάτες που επίσης σχετίζονται με τα ασφαλιστικά αυτά συμβόλαια και τυχόν παραβάσεις που έχουν συμβεί μεταξύ οδηγών και οχημάτων.

Τεχνικά, η βάση που περιλαμβάνει όλη την πληροφορία που προαναφέρθηκε έχει υλοποιηθεί στο σύστημα διαχείρισης βάσεων δεδομένων PostgreSQL και αποτελείται από έντεκα (11) πίνακες, των οποίων η περιγραφή θα γίνει παρακάτω. Όλοι αυτοί οι πίνακες σχετίζονται μεταξύ τους και περιλαμβάνουν έκαστος ένα διαφορετικό κομμάτι της πληροφορίας της βάσης.

1.2 Περιγραφή των σχέσεων της βάσης δεδομένων (ερώτημα 1α)

Οι έντεκα (11) πίνακες που συμπεριλαμβάνονται στην βάση δεδομένων είναι επιγραμματικά οι ακόλουθοι:

- vehicle_models
- infringements
- contracts
- clients_info
- drivers_info
- vehicles
- drivers
- infringements_drivers
- infringements_vehicles
- infringements_clients
- clients

Όσον αφορά τις πληροφορίες που διατηρούν κάθε μία από τις προαναφερθείσες σχέσεις, καθώς επίσης και τους περιορισμούς ακεραιότητας που διέπουν τους πίνακες αυτούς, αναφέρονται τα ακόλουθα:

1) vehicle_models

Ο πίνακας αυτός «κρατά» πληροφορίες για τα μοντέλα των αυτοκινήτων. Συγκεκριμένα, περιλαμβάνει τα ακόλουθα πεδία-χαρακτηριστικά: **model**, **manufacturer** και **category**. Κανένα από αυτά τα πεδία δεν επιτρέπεται να λάβει null τιμές. Το πρωτεύον κλειδί του πίνακα αυτού είναι το πεδίο **model**, αφού σχεδιαστικά κάθε μοντέλο εμφανίζεται μοναδικά στον πίνακα αυτό.

Ειδικότερα, στο πεδίο *model* αποθηκεύονται διάφορα μοντέλα οχημάτων. Στη συνέχεια, στο πεδίο *manufacturer* αποθηκεύονται κατασκευαστές οχημάτων, ενώ στο πεδίο *category* αποθηκεύεται η κατηγορία του οχήματος (car, truck ή motorcycle). Ο πίνακας αυτός δεν περιέχει ξένα κλειδιά. Για την δημιουργία της παρούσας σχέσης χρησιμοποιείται η ακόλουθη σύνταξη SQL:

```
CREATE TABLE Vehicle_Models
(
    model varchar(20) NOT NULL,
    manufacturer varchar(20) NOT NULL,
    category varchar(20) NOT NULL,
    PRIMARY KEY (model)
);
```

Τέλος, ακολουθεί ένα δείγμα των εγγραφών του πίνακα *vehicle_models*:

	model [PK] character varying (20)	manufacturer character varying (20)	category character varying (20)
1	Corsa	Opel	Car
2	FH12	Volvo	Truck
3	XT	Yamaha	Motorcycle
4	Yaris	Toyota	Car
5	R620	Scania	Truck
6	Hobby	Daytona	Motorcycle

2) infringements

Ο πίνακας αυτός συλλέγει τις πληροφορίες για τις παραβάσεις που μπορούν να συμβούν μεταξύ οδηγών και οχημάτων. Συγκεκριμένα, περιλαμβάνει τα ακόλουθα πεδία: **infringement_code**, **infringement_date**, **description** και **infringement_time**. Κανένα από αυτά τα πεδία δεν επιτρέπεται να λάβει null τιμές. Το πρωτεύον κλειδί του πίνακα αυτού είναι το πεδίο **infringement_code**, αφού κάθε παράβαση έχει και διαφορετικό κωδικό.

Ειδικότερα, στο πεδίο *infringement_code* αποθηκεύονται διάφοροι τυχαίοι κωδικοί παραβάσεων. Στη συνέχεια, στο πεδίο *infringement_date* αποθηκεύεται η ημερομηνία που έγινε η παράβαση, ενώ στο πεδίο *description* αποθηκεύεται μία σύντομη περιγραφή της παράβασης. Τέλος, το πεδίο *infringement_time* περιλαμβάνει την ώρα που συνέβη η παράβαση. Ο πίνακας αυτός δεν περιέχει ξένα κλειδιά. Για την δημιουργία της παρούσας σχέσης χρησιμοποιείται η ακόλουθη σύνταξη SQL:

```
CREATE TABLE Infringements
(
    infringement_code varchar(20) NOT NULL,
    infringement_date date NOT NULL,
    description text NOT NULL,
    infringement_time time without time zone NOT NULL,
    PRIMARY KEY (infringement_code)
);
```

Τέλος, ακολουθεί ένα δείγμα των εγγραφών του πίνακα *infringements*:

	infringement_code [PK] character varying (20)	infringement_date date	description text	infringement_time time without time zone
1	PO26887025_1000	2020-11-29	Accident in the city	07:18:00
2	PO26887025_1001	2020-02-04	Accident in the city	04:01:00
3	PO26887025_1002	2020-01-27	Serious damage	01:53:00
4	PO26887025_1003	2021-06-07	Illegal parking	04:01:00
5	PO26887025_1004	2020-11-04	Accident in the city	05:46:00
6	PO26887025_1005	2021-04-08	Serious damage	05:46:00

3) contracts

Ο πίνακας αυτός περιέχει τις πληροφορίες για τα συμβόλαια που έχει συνάψει η ασφαλιστική εταιρία με πελάτες και οδηγούς. Καθένα από τα συμβόλαια που περιέχονται στον πίνακα αυτό σχετίζεται με ένα όχημα.

Συγκεκριμένα, στην σχέση αυτή περιλαμβάνονται τα ακόλουθα πεδία: **contract_code**, **insurance_category**, **start_date**, **end_date** και **contract_cost**. Κανένα από αυτά τα πεδία δεν επιτρέπεται να λάβει null τιμές. Το πρωτεύον κλειδί του πίνακα αυτού είναι το πεδίο **contract_code**, διότι κάθε συμβόλαιο είναι διαφορετικό και, επομένως, έχει διαφορετικό κωδικό.

Ειδικότερα, στο πεδίο *contract_code* αποθηκεύονται διάφοροι τυχαίοι κωδικοί συμβολαίων. Στη συνέχεια, στο πεδίο *insurance_category* αποθηκεύεται μία εκ των τριών τιμές: *private*, *professional* και *mixed*. Στο πεδίο αυτό ουσιαστικά δηλώνεται η κατηγορία του συμβολαίου. Επίσης, τα πεδία *start_date* και *end_date* δηλώνουν την ημερομηνία έναρξης και την ημερομηνία λήξης του συμβολαίου αντίστοιχα. Τέλος, το πεδίο *contract_cost* περιέχει το κόστος του συμβολαίου. Ο πίνακας αυτός δεν περιέχει ξένα κλειδιά. Για την δημιουργία της παρούσας σχέσης χρησιμοποιείται η ακόλουθη σύνταξη SQL:

```
CREATE TABLE Contracts
(
    contract_code varchar(20) NOT NULL,
    insurance_category varchar(20) NOT NULL,
    start_date date NOT NULL,
    end_date date NOT NULL,
    contract_cost integer NOT NULL,
    PRIMARY KEY (contract_code)
);
```

Τέλος, ακολουθεί ένα δείγμα των εγγραφών του πίνακα *contracts*:

	contract_code [PK] character varying (20)	insurance_category character varying (20)	start_date date	end_date date	contract_cost integer
1	A087805	professional	2021-07-25	2025-08-30	42008
2	A975011	mixed	2021-06-20	2024-05-14	7117900
3	P599183	mixed	2014-06-15	2021-03-09	9935500
4	P217418	private	2021-07-28	2022-01-30	69106
5	P270842	professional	2009-07-24	2025-09-22	90195
6	P474722	mixed	2021-06-19	2022-05-19	4541600

4) clients_info

Ο πίνακας αυτός περιέχει τα στοιχεία των πελατών που σχετίζονται με ένα ή/και περισσότερα ασφαλιστικά συμβόλαια και κατ' επέκταση με ένα ή/και

περισσότερα οχήματα. Κάθε πελάτης που συμπεριλαμβάνεται σε αυτή τη σχέση, μπορεί να είναι (ή όχι) οδηγός του οχήματος με το οποίο συνδέεται.

Συγκεκριμένα, στην σχέση αυτή περιλαμβάνονται τα ακόλουθα πεδία: **first_name**, **last_name**, **gender**, **birth_date**, **license_number**, **phone_number1**, **phone_number2**, **email**, **street**, **zip_code**, **city** και **country**. Όλα τα πεδία, εκτός του **phone_number2**, δεν επιτρέπεται να λάβουν null τιμές. Το πρωτεύον κλειδί του πίνακα αυτού είναι το πεδίο **license_number**, καθώς κάθε πελάτης έχει διαφορετικό αριθμό διπλώματος. Τέλος, το πεδίο **email** είναι unique διότι θεωρούμε ότι κάθε πελάτης έχει διαφορετικό email.

Ειδικότερα, τα πεδία *first_name* και *last_name* αποθηκεύουν τα ονόματα και τα επίθετα των πελατών αντίστοιχα. Στη συνέχεια, στο πεδίο *gender* αποθηκεύεται το φύλο του πελάτη (male ή female). Επίσης, το πεδίο *birth_date* δηλώνει την ημερομηνία γέννησης του πελάτη. Επιπρόσθετα, το πεδίο *license_number* περιλαμβάνει τον αριθμό άδειας οδήγησης του πελάτη. Τα πεδία *phone_number1* και *phone_number2* περιέχουν τα τηλέφωνα επικοινωνίας με τους πελάτες (ένα κινητό και ένα σταθερό τηλέφωνο). Το πεδίο *phone_number2* μπορεί να επιδέχεται null τιμές, καθώς δεν είναι απαραίτητη η συμπλήρωση και του σταθερού και του κινητού τηλεφώνου του πελάτη. Το πεδίο *email* περιέχει το email του πελάτη και είναι unique καθώς κάθε πελάτης έχει διαφορετικό email. Τέλος, συμπεριλαμβάνεται και η διεύθυνση του πελάτη, η οποία έχει μοιραστεί στα πεδία *street*, *zip_code*, *city* και *country*. Διευκρίνιση: το πεδίο *street* περιέχει την οδό και τον αριθμό της οδού που κατοικεί ο πελάτης. Ο πίνακας αυτός δεν περιέχει ξένα κλειδιά. Για την δημιουργία της παρούσας σχέσης χρησιμοποιείται η ακόλουθη σύνταξη SQL:

```
CREATE TABLE Clients_Info
(
    first_name varchar(20) NOT NULL,
    last_name varchar(20) NOT NULL,
    gender varchar(20) NOT NULL,
    birth_date date NOT NULL,
    license_number bigint NOT NULL,
    phone_number1 varchar(20) NOT NULL,
    phone_number2 varchar(20),
    email varchar(30) NOT NULL UNIQUE,
    street varchar(20) NOT NULL,
    zip_code varchar(20) NOT NULL,
    city varchar(20) NOT NULL,
    country varchar(20) NOT NULL,
    PRIMARY KEY (license_number)
);
```

5) drivers info

Ο πίνακας αυτός περιέχει τα στοιχεία των οδηγών (όχι πελατών) που σχετίζονται με ένα ή/και περισσότερα ασφαλιστικά συμβόλαια και κατ'

επέκταση με ένα ή/και περισσότερα οχήματα. Συγκεκριμένα, στην σχέση αυτή περιλαμβάνονται τα ακόλουθα πεδία: **first_name**, **last_name**, **birth_date**, **street**, **zip_code**, **city**, **country**, **gender** και **license_number**. Όλα τα πεδία του πίνακα δεν επιτρέπεται να λάβουν null τιμές. Το πρωτεύον κλειδί του πίνακα αυτού είναι το πεδίο **license_number**, διότι κάθε οδηγός χαρακτηρίζεται από διαφορετικό αριθμό άδειας οδήγησης.

Ειδικότερα, τα πεδία *first_name* και *last_name* αποθηκεύουν τα ονόματα και τα επίθετα των οδηγών αντίστοιχα. Στη συνέχεια, στο πεδίο *gender* αποθηκεύεται το φύλο του οδηγού (male ή female). Επίσης, το πεδίο *birth_date* δηλώνει την ημερομηνία γέννησης του οδηγού και το πεδίο *license_number* περιλαμβάνει τον αριθμό άδειας οδήγησης του οδηγού. Τέλος, συμπεριλαμβάνεται και η διεύθυνση του οδηγού, η οποία έχει μοιραστεί στα πεδία *street*, *zip_code*, *city* και *country*. Διευκρίνιση: το πεδίο *street* περιέχει την οδό και τον αριθμό της οδού που κατοικεί ο οδηγός. Ο πίνακας αυτός δεν περιέχει ξένα κλειδιά. Για την δημιουργία της παρούσας σχέσης χρησιμοποιείται η ακόλουθη σύνταξη SQL:

```
CREATE TABLE Drivers_Info
(
    first_name varchar(20) NOT NULL,
    last_name varchar(20) NOT NULL,
    birth_date date NOT NULL,
    street varchar(20) NOT NULL,
    zip_code varchar(20) NOT NULL,
    city varchar(20) NOT NULL,
    country varchar(20) NOT NULL,
    gender varchar(20) NOT NULL,
    license_number bigint NOT NULL,
    PRIMARY KEY (license_number)
);
```

6) vehicles

Ο πίνακας αυτός περιέχει τα στοιχεία των οχημάτων που σχετίζονται με τα ασφαλιστικά συμβόλαια και κατ' επέκταση με έναν πελάτη και πολλούς οδηγούς. Συγκεκριμένα, στην σχέση αυτή περιλαμβάνονται τα ακόλουθα πεδία: **plate_number** (αριθμός πλαισίου του οχήματος), **car_license_number** (αριθμός άδειας κυκλοφορίας του οχήματος), **model** (το μοντέλο του οχήματος), **color** (το χρώμα του οχήματος), **first_year** (έτος πρώτης κυκλοφορίας του οχήματος), **current_price** (τρέχουσα αξία του οχήματος) και **contract_code** (κωδικός συμβολαίου με τον οποίο σχετίζεται το όχημα). Κανένα από αυτά τα πεδία δεν επιτρέπεται να λάβει null τιμές. Το πρωτεύον κλειδί του πίνακα αυτού είναι το πεδίο **plate_number** αφού ο αριθμός

πλασίου του οχήματος είναι μοναδικός για κάθε όχημα. Τέλος, τα πεδία *car_license_number* και *contract_code* έχουν δηλωθεί ως unique, καθώς ο αριθμός άδειας κυκλοφορίας και ο κωδικός συμβολαίου με τον οποίο σχετίζεται κάθε όχημα είναι μοναδικοί για κάθε όχημα.

Ο πίνακας αυτός περιλαμβάνει το πεδίο *model* που αναφέρεται (ξένο κλειδί) στη σχέση **vehicle_models**, διότι κάθε μοντέλο ενός οχήματος οφείλεται να αναγράφεται στον πίνακα *vehicle_models* που συλλέγει τις πληροφορίες για όλα τα μοντέλα των οχημάτων. Ομοίως, το πεδίο *contract_code* του πίνακα αναφέρεται στη σχέση **contracts**.

Για την δημιουργία της παρούσας σχέσης χρησιμοποιείται η ακόλουθη σύνταξη SQL:

```
CREATE TABLE Vehicles
(
    plate_number varchar(20) NOT NULL,
    car_license_number varchar(20) NOT NULL UNIQUE,
    model varchar(20) NOT NULL,
    color varchar(20) NOT NULL,
    first_year integer NOT NULL,
    current_price integer NOT NULL,
    contract_code varchar(20) NOT NULL UNIQUE,
    PRIMARY KEY (plate_number),
    FOREIGN KEY (model) references Vehicle_Models ON DELETE CASCADE,
    FOREIGN KEY (contract_code) references Contracts ON DELETE CASCADE
);
```

7) drivers

Ο πίνακας αυτός περιέχει όλα τα συμβόλαια με τα οποία μπορεί να σχετίζεται ένας οδηγός (όχι πελάτης). Συγκεκριμένα, στην σχέση αυτή περιλαμβάνονται τα ακόλουθα πεδία: **contract_code** (κωδικός συμβολαίου με το οποίο σχετίζεται ο οδηγός) και **license_number** (αριθμός άδειας οδήγησης του οδηγού). Κανένα από αυτά τα πεδία δεν επιτρέπεται να λάβει null τιμές. Το πρωτεύον κλειδί του πίνακα αυτού είναι ο **συνδυασμός** των δύο παραπάνω πεδίων, καθώς θεωρούμε ότι ένας οδηγός μπορεί να σχετίζεται με πολλά συμβόλαια. Ανάλογα, ένα συμβόλαιο μπορεί να σχετίζεται με πολλούς οδηγούς. Άρα, τόσο το πεδίο *contract_code* όσο και το *license_number*, μπορούν να επαναλαμβάνονται στη σχέση αυτή.

Ο πίνακας αυτός περιλαμβάνει το πεδίο *contract_code* που αναφέρεται (ξένο κλειδί) στη σχέση **contracts**, διότι κάθε κωδικός συμβολαίου ορίζει μοναδικά ένα συμβόλαιο, το οποίο πρέπει να είναι καταχωρημένο στον πίνακα

contracts που περιέχει πληροφορίες για όλα τα συμβόλαια. Ομοίως, το πεδίο *license_number* του πίνακα αναφέρεται στη σχέση **drivers_info**, καθώς ο πίνακας αυτός περιέχει τις πληροφορίες για όλους τους οδηγούς που περιλαμβάνει η βάση, άρα κάθε αριθμός άδειας οδήγησης που αντιστοιχεί μοναδικά σε έναν οδηγό, οφείλεται να υπάρχει στον πίνακα *drivers_info*.

Για την δημιουργία της παρούσας σχέσης χρησιμοποιείται η ακόλουθη σύνταξη SQL:

```
CREATE TABLE Drivers
(
    contract_code varchar(20) NOT NULL,
    license_number bigint NOT NULL,
    PRIMARY KEY (contract_code, license_number),
    FOREIGN KEY (contract_code) references Contracts ON DELETE CASCADE,
    FOREIGN KEY (license_number) references Drivers_Info ON DELETE CASCADE
);
```

Τέλος, ακολουθεί ένα δείγμα των εγγραφών του πίνακα *drivers*:

	contract_code [PK] character varying (20)	license_number [PK] bigint
1	A087805	7371415
2	A975011	9546096
3	P599183	6628274
4	P217418	5020179
5	P270842	379687
6	A087805	4585079

8) clients

Ο πίνακας αυτός περιέχει όλα τα συμβόλαια με τα οποία μπορεί να σχετίζεται ένας πελάτης. Συγκεκριμένα, στην σχέση αυτή περιλαμβάνονται τα ακόλουθα πεδία: **contract_code** (κωδικός συμβολαίου με το οποίο σχετίζεται ο πελάτης), **license_number** (αριθμός άδειας οδήγησης του πελάτη) και το πεδίο **isDriver** που λαμβάνει λογικές τιμές. Αν το πεδίο αυτό έχει την τιμή *true*, τότε αυτό σημαίνει ότι ο συγκεκριμένος πελάτης είναι και οδηγός του οχήματος με το οποίο σχετίζεται. Σε κάθε άλλη περίπτωση, το πεδίο *isDriver* λαμβάνει την τιμή *false*.

Κανένα από αυτά τα πεδία δεν επιτρέπεται να λάβει null τιμές. Το πρωτεύον κλειδί του πίνακα αυτού είναι το πεδίο **contract_code**, αφού ένα συμβόλαιο σχετίζεται με ένα μόνο πελάτη

Ο πίνακας αυτός περιλαμβάνει το πεδίο *contract_code* που αναφέρεται (ξένο κλειδί) στη σχέση **contracts**, διότι κάθε κωδικός συμβολαίου ορίζει μοναδικά ένα συμβόλαιο, το οποίο πρέπει να είναι καταχωρημένο στον πίνακα *contracts* που περιέχει πληροφορίες για όλα τα συμβόλαια. Ομοίως, το πεδίο *license_number* του πίνακα αναφέρεται στη σχέση **clients_info**, καθώς ο πίνακας αυτός περιέχει τις πληροφορίες για όλους τους πελάτες που περιλαμβάνει η βάση, άρα κάθε αριθμός άδειας οδήγησης που αντιστοιχεί μοναδικά σε έναν πελάτη, οφείλεται να υπάρχει στον πίνακα *clients_info*.

Για την δημιουργία της παρούσας σχέσης χρησιμοποιείται η ακόλουθη σύνταξη SQL:

```
CREATE TABLE Clients
(
    license_number bigint NOT NULL,
    contract_code varchar(20) NOT NULL,
    "isDriver" boolean NOT NULL,
    PRIMARY KEY (contract_code),
    FOREIGN KEY (license_number) references Clients_Info ON DELETE CASCADE,
    FOREIGN KEY (contract_code) references Contracts ON DELETE CASCADE
);
```

Τέλος, ακολουθεί ένα δείγμα των εγγραφών του πίνακα *clients*:

	license_number bigint	contract_code [PK] character varying (20)	isDriver boolean
1	7307866	A087805	false
2	1373024	A975011	false
3	3657661	P599183	true
4	3657661	P217418	false
5	3657661	P270842	false
6	8288372	P474722	true

9) infringements_drivers

Ο πίνακας αυτός περιέχει τις πληροφορίες για όλες τις παραβάσεις που τυχόν έχει υποπέσει ένας οδηγός (όχι πελάτης). Συγκεκριμένα, ο πίνακας αυτός περιλαμβάνει τα πεδία **infringement_code** (δηλώνει τον κωδικό

παράβασης) και **license_number** (δηλώνει τον αριθμό άδειας οδήγησης του οδηγού). Κανένα από αυτά τα δύο πεδία δεν επιδέχεται null τιμές. Το πρωτεύον κλειδί του πίνακα είναι ο **συνδυασμός** των δύο παραπάνω πεδίων, καθώς θεωρούμε ότι ένας οδηγός μπορεί να έχει υποπέσει σε πολλές παραβάσεις. Αντιστοίχως, σε μία παράβαση μπορεί να έχουν εμπλακεί πολλοί οδηγοί.

Όσον αφορά τα ξένα κλειδιά, το πεδίο *infringement_code* αναφέρεται στη σχέση **infringements**, αφού κάθε κωδικός παράβασης πρέπει να έχει καταγραφεί στον πίνακα *infringements* που περιέχει πληροφορίες για όλες τις παραβάσεις που έχουν συμβεί. Επίσης, το πεδίο *license_number* αναφέρεται στη σχέση **drivers_info**, καθώς ο πίνακας αυτός περιέχει τις πληροφορίες για όλους τους οδηγούς που περιλαμβάνει η βάση, άρα κάθε αριθμός άδειας οδήγησης που αντιστοιχεί μοναδικά σε έναν οδηγό, οφείλεται να υπάρχει στον πίνακα *drivers_info*.

Για την δημιουργία της παρούσας σχέσης χρησιμοποιείται η ακόλουθη σύνταξη SQL:

```
CREATE TABLE Infringements_Drivers
(
    infringement_code varchar(20) NOT NULL,
    license_number bigint NOT NULL,
    PRIMARY KEY (infringement_code, license_number),
    FOREIGN KEY (infringement_code) references Infringements ON DELETE CASCADE,
    FOREIGN KEY (license_number) references Drivers_Info ON DELETE CASCADE
);
```

Τέλος, ακολουθεί ένα δείγμα των εγγραφών του πίνακα *infringements_drivers*:

	infringement_code [PK] character varying (20)	license_number [PK] bigint
1	PO26887025_1000	820081
2	PO26887025_1001	3492248
3	PO26887025_1002	8006023
4	PO26887025_1003	8124154
5	PO26887025_1004	9546096
6	PO26887025_1005	8236767

10) infringements_clients

Ο πίνακας αυτός περιέχει τις πληροφορίες για όλες τις παραβάσεις που τυχόν έχει υποπέσει ένας πελάτης. Συγκεκριμένα, ο πίνακας αυτός

περιλαμβάνει τα πεδία **infringement_code** (δηλώνει τον κωδικό παράβασης) και **license_number** (δηλώνει τον αριθμό άδειας οδήγησης του πελάτη). Κανένα από αυτά τα δύο πεδία δεν επιδέχεται null τιμές. Το πρωτεύον κλειδί του πίνακα είναι ο **συνδυασμός** των δύο παραπάνω πεδίων, καθώς θεωρούμε ότι ένας πελάτης μπορεί να έχει υποπέσει σε πολλές παραβάσεις, αν ο ίδιος οδηγεί το όχημα με το οποίο σχετίζεται. Αντιστοίχως, σε μία παράβαση μπορεί να έχουν εμπλακεί πολλοί πελάτες.

Όσον αφορά τα ξένα κλειδιά, το πεδίο *infringement_code* αναφέρεται στη σχέση **infringements**, αφού κάθε κωδικός παράβασης πρέπει να έχει καταγραφεί στον πίνακα *infringements* που περιέχει πληροφορίες για όλες τις παραβάσεις που έχουν συμβεί. Επίσης, το πεδίο *license_number* αναφέρεται στη σχέση **clients_info**, καθώς ο πίνακας αυτός περιέχει τις πληροφορίες για όλους τους πελάτες που περιλαμβάνει η βάση, άρα κάθε αριθμός άδειας οδήγησης που αντιστοιχεί μοναδικά σε έναν πελάτη, οφείλεται να υπάρχει στον πίνακα *clients_info*.

Για την δημιουργία της παρούσας σχέσης χρησιμοποιείται η ακόλουθη σύνταξη SQL:

```
CREATE TABLE Infringements_Clients
(
    infringement_code varchar(20) NOT NULL,
    license_number bigint NOT NULL,
    PRIMARY KEY (infringement_code, license_number),
    FOREIGN KEY (infringement_code) references Infringements ON DELETE CASCADE,
    FOREIGN KEY (license_number) references Clients_Info ON DELETE CASCADE
);
```

Τέλος, ακολουθεί ένα δείγμα των εγγραφών του πίνακα *infringements_clients*:

	infringement_code [PK] character varying (20)	license_number [PK] bigint
1	PO26887025_1000	1074911
2	PO26887025_1002	9170944
3	PO26887025_1004	9640655
4	PO26887025_1006	1422054
5	PO26887025_1008	6961770
6	PO26887025_1010	2865818

11) infringements_vehicles

Ο πίνακας αυτός περιέχει τις πληροφορίες για όλες τις παραβάσεις που τυχόν συμμετέχει ένα όχημα. Συγκεκριμένα, ο πίνακας αυτός περιλαμβάνει τα πεδία **infringement_code** (δηλώνει τον κωδικό παράβασης) και **plate_number** (δηλώνει τον αριθμό πλαισίου του οχήματος). Κανένα από αυτά τα δύο πεδία δεν επιδέχεται null τιμές. Το πρωτεύον κλειδί του πίνακα είναι ο **συνδυασμός** των δύο παραπάνω πεδίων, καθώς θεωρούμε ότι ένα όχημα μπορεί να συμμετέχει σε πολλές παραβάσεις. Αντιστοίχως, σε μία παράβαση μπορεί να έχουν εμπλακεί πολλά οχήματα.

Όσον αφορά τα ξένα κλειδιά, το πεδίο *infringement_code* αναφέρεται στη σχέση **infringements**, αφού κάθε κωδικός παράβασης πρέπει να έχει καταγραφεί στον πίνακα *infringements* που περιέχει πληροφορίες για όλες τις παραβάσεις που έχουν συμβεί. Επίσης, το πεδίο *plate_number* αναφέρεται στη σχέση **vehicles**, καθώς ο πίνακας αυτός περιέχει τις πληροφορίες για όλα τα οχήματα που περιλαμβάνει η βάση, άρα κάθε αριθμός πλαισίου που αντιστοιχεί μοναδικά σε ένα όχημα, οφείλεται να υπάρχει στον πίνακα *vehicles*.

Για την δημιουργία της παρούσας σχέσης χρησιμοποιείται η ακόλουθη σύνταξη SQL:

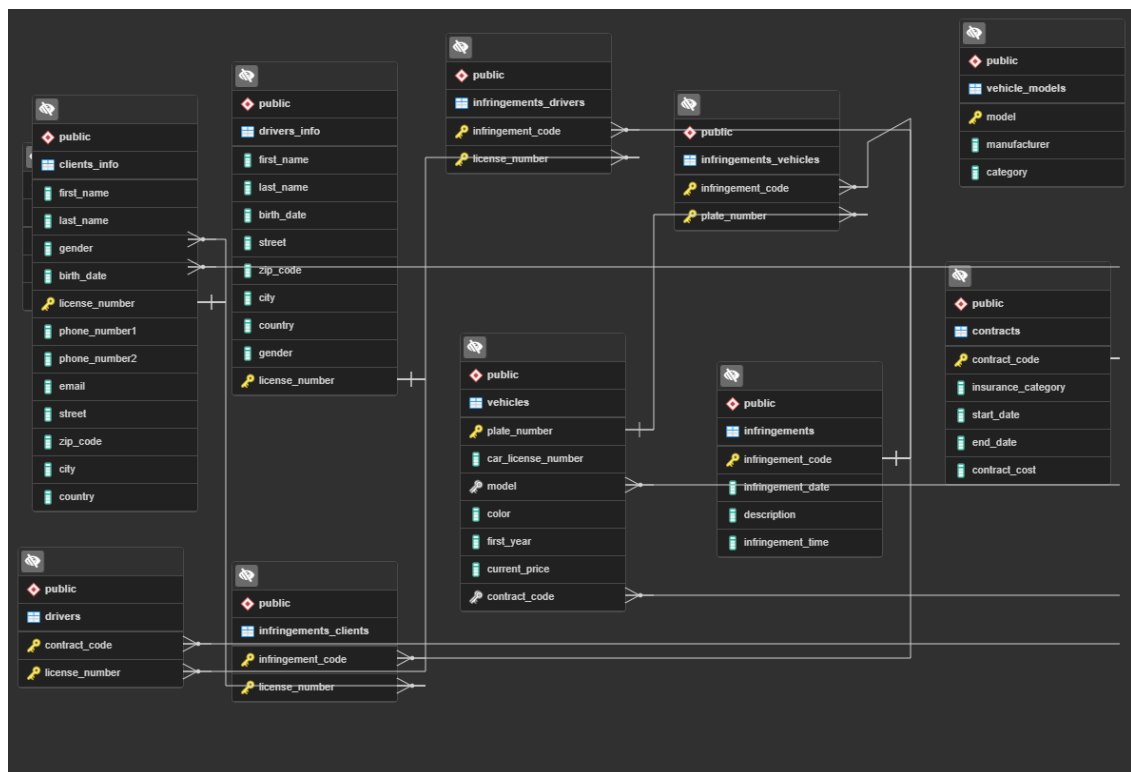
```
CREATE TABLE Infringements_Vehicles
(
    infringement_code varchar(20) NOT NULL,
    plate_number varchar(20) NOT NULL,
    PRIMARY KEY (infringement_code, plate_number),
    FOREIGN KEY (infringement_code) references Infringements ON DELETE CASCADE,
    FOREIGN KEY (plate_number) references Vehicles ON DELETE CASCADE
);
```

Τέλος, ακολουθεί ένα δείγμα των εγγραφών του πίνακα *infringements_vehicles*:

	infringement_code [PK] character varying (20)	plate_number [PK] character varying (20)
1	PO26887025_1001	3468SV
2	PO26887025_1002	8932YU
3	PO26887025_1003	9116FF
4	PO26887025_1004	2222CP
5	PO26887025_1005	9166RF
6	PO26887025_1006	2974EX

1.3 Το διάγραμμα ERD της βάσης δεδομένων (ερώτημα 1α)

Παρακάτω, φαίνεται το διάγραμμα της σχεσιακής βάσης δεδομένων που δημιουργήθηκε. Σε αυτό περιλαμβάνονται και οι έντεκα (11) πίνακες που περιεγράφηκαν, καθώς και οι μεταξύ τους σχέσεις (περιορισμοί ακεραιότητας οντότητας και αναφορικής ακεραιότητας):



Αξίζει να σημειωθεί ότι το παραπάνω διάγραμμα είναι αποθηκευμένο ως εικόνα στον φάκελο «Ερώτημα 1» του project.

1.4 Κανονικοποίηση της βάσης δεδομένων (ερώτημα 1b)

Η σχεσιακή βάση δεδομένων που δημιουργήθηκε, έχει όλες τις σχέσεις της σε BCNF. Απόδειξη:

1) vehicle_models (model, manufacturer, category)

Στη σχέση αυτή υπάρχουν οι συναρτησιακές εξαρτήσεις:

model → *manufacturer,category*

Ένα μοντέλο προσδιορίζει τον κατασκευαστή και την κατηγορία του οχήματος. Ωστόσο, ο κατασκευαστής του οχήματος και η κατηγορία του οχήματος, δεν προσδιορίζουν κανένα από τα υπόλοιπα πεδία της σχέσης. Άρα, έχουμε υποψήφιο κλειδί το πεδίο *model*, το οποίο γίνεται πρωτεύον κλειδί της σχέσης. Επομένως, η σχέση είναι σε BCNF.

2) infringements (infringement_code, infringement_date, description, infringement_time)

Στη σχέση αυτή υπάρχουν οι συναρτησιακές εξαρτήσεις:

infringement_code → *infringement_date, description, infringement_time*

Ο κωδικός μίας παράβασης προσδιορίζει όλα τα άλλα πεδία του πίνακα, καθώς υπάρχει μοναδικός κωδικός για κάθε μία παράβαση. Δεν υπάρχουν άλλες εξαρτήσεις. Άρα, το πεδίο *infringement_code* είναι υποψήφιο κλειδί, άρα και πρωτεύον κλειδί της σχέσης *infringements*. Επομένως, η σχέση βρίσκεται σε BCNF.

3) contracts (contract_code, insurance_category, start_date, end_date, contract_cost)

Στη σχέση αυτή υπάρχουν οι συναρτησιακές εξαρτήσεις:

contract_code → *insurance_category, start_date, end_date, contract_cost*

Ο κωδικός ενός συμβολαίου είναι μοναδικός για κάθε συμβόλαιο. Για αυτό τον λόγο ισχύει η παραπάνω συναρτησιακή εξάρτηση. Δεν υπάρχουν άλλες εξαρτήσεις. Άρα, το πεδίο *contract_code* είναι υποψήφιο κλειδί, άρα και πρωτεύον κλειδί της σχέσης *contracts*. Επομένως, η σχέση βρίσκεται σε BCNF.

4) clients_info (first_name, last_name, gender, birth_date, license_number, phone_number1, phone_number2, email, street, zip_code, city, country)

Στη σχέση αυτή υπάρχουν οι συναρτησιακές εξαρτήσεις:

license_number → *first_name, last_name, gender, birth_date, phone_number1, phone_number2, email, street, zip_code, city, country*

email → *first_name, last_name, gender, birth_date, phone_number1, phone_number2, license_number, street, zip_code, city, country*

Θεωρούμε ότι το email και ο αριθμός άδειας οδήγησης κάθε πελάτη είναι μοναδικά για κάθε πελάτη στη βάση. Δεν υπάρχουν άλλες εξαρτήσεις. Άρα, τα υποψήφια κλειδιά της σχέσης είναι α) *license_number* και β) *email*. Από αυτά, θεωρούμε ως πρωτεύον κλειδί το πεδίο *license_number*, ενώ το πεδίο *email* το ορίζουμε ως unique. Επομένως, η σχέση βρίσκεται σε BCNF.

5) drivers_info (first_name, last_name, birth_date, street, zip_code, city, country, gender, license_number)

Στη σχέση αυτή υπάρχουν οι συναρτησιακές εξαρτήσεις:

license_number → *first_name, last_name, gender, birth_date, street, zip_code, city, country*

Θεωρούμε ότι ο αριθμός άδειας οδήγησης κάθε οδηγού είναι μοναδικός για κάθε οδηγό στη βάση. Δεν υπάρχουν άλλες εξαρτήσεις. Άρα, το υποψήφιο κλειδί της σχέσης είναι το πεδίο *license_number*. Επομένως, το πεδίο *license_number* είναι πρωτεύον κλειδί της σχέσης *drivers_info*. Επομένως, η σχέση βρίσκεται σε BCNF.

6) vehicles (plate_number, car_license_number, model, color, first_year, current_price, contract_code)

Στη σχέση αυτή υπάρχουν οι συναρτησιακές εξαρτήσεις:

plate_number → *car_license_number, model, color, first_year, current_price, contract_code*

car_license_number → *plate_number, model, color, first_year, current_price, contract_code*

contract_code → *car_license_number*, *model*, *color*, *first_year*, *current_price*, *plate_number*

Οι τρεις (3) παραπάνω συναρτησιακές εξαρτήσεις ισχύουν καθώς κάθε όχημα έχει έναν διαφορετικό αριθμό πλαισίου (*plate_number*). Άρα, το πεδίο *plate_number* προσδιορίζει μοναδικά μία εγγραφή στον πίνακα *vehicles*. Ομοίως, τα ίδια ισχύουν και για το πεδίο *car_license_number*. Τέλος, για το πεδίο *contract_code* θεωρούμε ότι κάθε κωδικός συμβολαίου σχετίζεται με ένα μόνο όχημα, επομένως, αυτό το πεδίο δεν μπορεί να λαμβάνει διπλότυπες τιμές στη σχέση *vehicles*.

Συνεπώς, τα τρία υποψήφια κλειδιά της σχέσης *vehicles* είναι α) *plate_number*, β) *car_license_number* και γ) *contract_code*. Ορίζουμε ως πρωτεύον κλειδί του πίνακα το πεδίο *plate_number*, ενώ τα υπόλοιπα δύο υποψήφια κλειδιά τα δηλώνουμε ως *unique*.

7) drivers (contract_code, license_number)

Στη σχέση αυτή δεν υπάρχουν συναρτησιακές εξαρτήσεις. Αυτό συμβαίνει διότι οι πληροφορίες που αποθηκεύονται εδώ είναι όλα τα συμβόλαια που σχετίζονται με οδηγούς (όχι πελάτες). Επομένως, ένα συμβόλαιο δεν προσδιορίζει μονοσήμαντα έναν μόνο οδηγό, αφού για το ίδιο συμβόλαιο, μπορούν να έχουν οριστεί πολλοί οδηγοί. Επίσης, ένας οδηγός (προσδιορίζεται μοναδικά από το πεδίο *license_number*) δεν προσδιορίζει μοναδικά έναν μόνο κωδικό συμβολαίου (και κατ' επέκταση ένα συμβόλαιο). Δηλαδή, υπάρχει πάντα η περίπτωση ένας οδηγός να σχετίζεται με πολλά συμβόλαια.

Επομένως, ο πίνακας έχει ως πρωτεύον κλειδί τον συνδυασμό των δύο αυτών πεδίων. Άρα, η σχέση βρίσκεται σε BCNF.

8) clients (contract_code, license_number, isDriver)

Στην σχέση αυτή υπάρχουν οι ακόλουθες συναρτησιακές εξαρτήσεις:

contract_code → *license_number*, *isDriver*

Δεν υπάρχουν άλλες συναρτησιακές εξαρτήσεις. Το πεδίο *contract_code* προσδιορίζει όλα τα υπόλοιπα πεδία της σχέσης, αφού κάθε συμβόλαιο προσδιορίζεται μοναδικά από έναν μόνο πελάτη. Ωστόσο, ένας πελάτης

μπορεί να έχει συνάψει και άλλα συμβόλαια με την ασφαλιστική εταιρία. Επομένως, το υποψήφιο κλειδί, και άρα πρωτεύον κλειδί, της σχέσης είναι το πεδίο *contract_code*. Η σχέση βρίσκεται σε BCNF.

9) *infringements_drivers* (infringement_code, license_number)

Στη σχέση αυτή δεν υπάρχουν συναρτησιακές εξαρτήσεις. Ο πίνακας αυτός συλλέγει τις πληροφορίες για όλες τις παραβάσεις που έχουν υποπέσει οι οδηγοί (όχι πελάτες). Επομένως, λόγω του γεγονότος ότι ένας οδηγός μπορεί να υποπέσει σε πολλές παραβάσεις και μία παράβαση μπορεί να περιλαμβάνει πολλούς οδηγούς, θεωρούμε ως υποψήφιο κλειδί της σχέσης τον συνδυασμό των πεδίων *infringement_code* και *license_number*. Άρα, το πρωτεύον κλειδί του πίνακα είναι και τα δύο πεδία αυτού. Ο πίνακας βρίσκεται σε BCNF.

10) *infringements_clients* (infringement_code, license_number)

Στη σχέση αυτή δεν υπάρχουν συναρτησιακές εξαρτήσεις. Ο πίνακας αυτός συλλέγει τις πληροφορίες για όλες τις παραβάσεις που έχουν υποπέσει οι πελάτες που είναι και οδηγοί των οχημάτων με τα οποία σχετίζονται. Επομένως, λόγω του γεγονότος ότι ένας πελάτης-οδηγός μπορεί να υποπέσει σε πολλές παραβάσεις και μία παράβαση μπορεί να περιλαμβάνει πολλούς πελάτες-οδηγούς, θεωρούμε ως υποψήφιο κλειδί της σχέσης τον συνδυασμό των πεδίων *infringement_code* και *license_number*. Άρα, το πρωτεύον κλειδί του πίνακα είναι και τα δύο πεδία αυτού. Ο πίνακας βρίσκεται σε BCNF.

11) *infringements_vehicles* (infringement_code, plate_number)

Στη σχέση αυτή δεν υπάρχουν συναρτησιακές εξαρτήσεις. Ο πίνακας αυτός συλλέγει τις πληροφορίες για όλα τα οχήματα που συμμετέχουν σε παράβαση. Επομένως, λόγω του γεγονότος ότι ένα όχημα μπορεί να συμμετέχει σε πολλές παραβάσεις και μία παράβαση μπορεί να περιλαμβάνει πολλά οχήματα, θεωρούμε ως υποψήφιο κλειδί της σχέσης τον συνδυασμό των πεδίων *infringement_code* και *plate_number*. Άρα, το πρωτεύον κλειδί του πίνακα είναι και τα δύο πεδία αυτού. Ο πίνακας βρίσκεται σε BCNF.

2 Εντολές SELECT (ερώτημα 2)

Ερώτημα 2α: Ποια (νέα) συμβόλαια υπεγράφησαν τον τελευταίο μήνα και ποιοι είναι οι πελάτες και οι οδηγοί που σχετίζονται με αυτά.

```
-- Ερώτημα 2α
-- βρες όλους τους οδηγούς που σχετίζονται με συμβόλαια
-- τα οποία υπεγράφησαν τον τελευταίο μήνα
(
  SELECT first_name,last_name,contract_code,'Driver' AS Category
  FROM contracts NATURAL JOIN drivers NATURAL JOIN drivers_info
  WHERE contract_code IN
    (
      -- Νέα συμβόλαια
      SELECT contract_code
      FROM contracts
      WHERE EXTRACT (YEAR FROM start_date)=(SELECT date_part('year',CURRENT_DATE)) AND EXTRACT (MONTH FROM start_date)=(SELECT
date_part('month',CURRENT_DATE))
    )
)
UNION
--βρες όλους τους πελάτες που σχετίζονται με συμβόλαια
-- τα οποία υπεγράφησαν τον τελευταίο μήνα
(
  SELECT first_name,last_name,contract_code,'Client' AS Category
  FROM contracts NATURAL JOIN clients NATURAL JOIN clients_info
  WHERE contract_code IN
    (
      -- Νέα συμβόλαια
      SELECT contract_code
      FROM contracts
      WHERE EXTRACT (YEAR FROM start_date)=(SELECT date_part('year',CURRENT_DATE)) AND EXTRACT (MONTH FROM start_date)=(SELECT
date_part('month',CURRENT_DATE))
    )
)
ORDER BY contract_code
```

Στο συγκεκριμένο ερώτημα ψάχνουμε τα συμβόλαια που υπεγράφησαν τον τελευταίο μήνα καθώς και τους οδηγούς και τους πελάτες που σχετίζονται με αυτά. Αρχικά συνδέουμε με NATURAL JOIN τους πίνακες *contracts*, *drivers*, *drivers_info* και από αυτούς αναζητούμε το όνομα, το επίθετο και τον κωδικό συμβολαίου των οδηγών, των οποίων το συμβόλαιο υπογράφηκε το τρέχον έτος και τον τρέχοντα μήνα. Αυτό επιτυγχάνεται με την σύγκριση τιμών του CURRENT_DATE (τρέχουσα ημερομηνία) και start_date (ημερομηνία υπογραφής συμβολαίου) και με τη βοήθεια της εντολής EXTRACT, μέσω της οποίας παίρνουμε την χρονιά και τον μήνα. Επίσης προσθέτουμε την στήλη “category” ώστε να εμφανίζεται η ιδιότητα του ατόμου (Driver).

Ομοίως συνδέουμε τους πίνακες *contracts*, *clients* και *clients_info* για να βρούμε τις αντίστοιχες πληροφορίες για τους πελάτες και αυτήν την φορά στην στήλη “category” εμφανίζεται η ιδιότητα “Client”. Η σύνδεση των στοιχείων που προκύπτουν γίνεται με την εντολή UNION. Τέλος κατατάσσουμε τα αποτελέσματα με κριτήριο τον κωδικό συμβολαίου.

Αποτέλεσμα:

	first_name character varying (20)	last_name character varying (20)	contract_code character varying (20)	category text
1	Patricia	Orcutt	A484319	Driver
2	Manuel	Ladaille	A484319	Client
3	Erik	Chapman	A484319	Driver
4	Jose	Wood	A647907	Driver
5	Ruth	Zimmerman	A647907	Client
6	Charlie	Nelson	A975011	Client
7	Giel	Watson	A975011	Driver
8	Daan	Pensec	A975011	Driver
9	Marcin	Knight	P209751	Client
10	Eleanor	Linhart	P209751	Driver
11	Megan	Pierce	P270842	Driver
12	Marta	Wakefield	P270842	Client
13	Marty	Poole	P270842	Driver
14	Teun	Gibson	P329398	Driver
15	Erik	Chapman	P329398	Driver
16	Megan	Hankins	P329398	Client
17	Megan	Pierce	P329398	Driver
18	Sally	DelRosso	P474722	Client
19	Al	Guyer	P474722	Driver
20	Peg	Petrzelka	P590699	Driver
21	Bob	King	P590699	Client
22	Duncan	Weaver	P671261	Client
23	Ed	Weaver	P671261	Driver
24	Shermie	Wilson	P671261	Driver
25	Sylvia	Gibson	P774736	Driver
26	Jonas	Overton	P774736	Client
27	Edwina	Stewart	P811636	Driver
28	Stanislaw	Thompson	P811636	Client

Ερώτημα 2b: Ποια συμβόλαια αναμένεται να λήξουν τον επόμενο μήνα και ποια είναι τα τηλέφωνα επικοινωνίας των πελατών που σχετίζονται με αυτά.

```
-- Ερώτημα 2b
-- Παραδοχή: θεωρούμε ότι η τρέχουσα ημερομηνία είναι Ιούνιος του 2021 (06/2021)
SELECT contract_code, phone_number1, phone_number2
FROM contracts natural join clients natural join clients_info
WHERE contract_code IN
(
    SELECT contract_code
    FROM contracts
    WHERE EXTRACT (YEAR FROM end_date)=2021 AND EXTRACT (MONTH FROM end_date)=7
)
```

Στο ερώτημα αυτό ψάχνουμε τα συμβόλαια που αναμένεται να λήξουν τον επόμενο μήνα και τα τηλέφωνα επικοινωνίας των πελατών που σχετίζονται με αυτά. Παραδοχή: για το συγκεκριμένο ερώτημα θεωρούμε ότι η τρέχουσα ημερομηνία είναι ο Ιούνιος του 2021 (06/2021).

Αρχικά, συνδέουμε με NATURAL JOIN τους πίνακες *contracts*, *clients* και *clients_info* έτσι ώστε να πάρουμε τα πεδία: κωδικός συμβολαίου (*contract_code*) και τα τηλέφωνα των πελατών (*phone_number1*, *phone_number2*), ο οποίος κωδικός συμβολαίου αναμένεται να λήξει τον επόμενο μήνα δηλαδή τον Ιούλιο του 2021. Αυτό επιτυγχάνεται μέσω του WHERE που μας επιτρέπει να επιλέξουμε τον συγκεκριμένο κωδικό συμβολαίου από τον πίνακα *contracts* στον οποίο συγκρίνουμε αν ο χρόνος και ο μήνας της λήξης της ημερομηνίας του συμβολαίου ισούται με το 2021 και το επτά (7) αντίστοιχα, μέσω από την εντολή EXTRACT που μας επιτρέπει να πάρουμε τον χρόνο και μήνα μέσα από το πεδίο *end_date*.

Αποτέλεσμα:

	contract_code character varying (20)	phone_number1 character varying (20)	phone_number2 character varying (20)
1	A647907	204-3388	621-5513
2	A925510	807-971-2943	1-252-887-9180
3	A298941	1-416-443-3339	1-551-5610
4	A701776	1-322-270-2720	865-6349
5	P914992	1-030-971-1181	1-642-9174
6	A625101	1-689-5740	[null]

Ερώτημα 2c: Ποιος είναι ο αριθμός των συμβολαίων που υπεγράφησαν (παραλλαγή: που δεν ανανεώθηκαν) ανά ασφαλιστική κατηγορία και ανά έτος για την πενταετία 2016-2020.

```
-- Ερώτημα 2c
SELECT COUNT(DISTINCT contract_code) AS number_of_contracts, insurance_category, EXTRACT (YEAR FROM start_date) AS contract_year
FROM contracts
WHERE EXTRACT (YEAR FROM start_date)>=2016 AND EXTRACT (YEAR FROM start_date)<=2020
GROUP BY insurance_category,EXTRACT (YEAR FROM start_date)
```

Σε αυτό το ερώτημα αναζητούμε τον αριθμό συμβολαίων που υπεγράφησαν ανά ασφαλιστική κατηγορία και ανά έτος για την πενταετία 2016-2020. Αρχικά, επιλέγουμε από τον πίνακα *contracts*: το πλήθος των κωδικών συμβολαίων (μέσω της COUNT), το πεδίο της ασφαλιστικής κατηγορίας και την χρονιά που υπεγράφη το συμβόλαιο, δηλαδή κάνουμε EXTRACT τον χρόνο από το πεδίο *start_date*. Για να βρούμε τον αριθμό συμβολαίων χρησιμοποιούμε το WHERE μέσα από το οποίο συγκρίνουμε την χρονιά που υπεγράφη το συμβόλαιο αν ανήκει στην πενταετία 2016-2020. Τέλος με το GROUP BY κατατάσσουμε τα αποτελέσματα ανά ασφαλιστική κατηγορία και ανά την χρονιά που υπεγράφη το συμβόλαιο.

Αποτέλεσμα:

	number_of_contracts bigint	insurance_category character varying (20)	contract_year double precision
1	2	mixed	2016
2	4	mixed	2017
3	2	mixed	2018
4	1	mixed	2019
5	1	mixed	2020
6	1	private	2016
7	1	private	2017
8	1	private	2019
9	2	professional	2016
10	2	professional	2017
11	3	professional	2018
12	2	professional	2019
13	1	professional	2020

Ερώτημα 2c (Παραλλαγή):

```
-- Ερώτημα 2c (Παραλλαγή)
SELECT COUNT(DISTINCT contract_code) AS number_of_contracts, insurance_category, EXTRACT (YEAR FROM end_date) AS expired_year
FROM contracts
WHERE (EXTRACT (YEAR FROM end_date)>=2016 AND EXTRACT (YEAR FROM end_date)<=2020) AND (EXTRACT (YEAR FROM start_date)<2016 OR EXTRACT (YEAR FROM start_date)>2020)
GROUP BY insurance_category,EXTRACT (YEAR FROM end_date)
```

Σε αυτό το ερώτημα ψάχνουμε πόσα συμβόλαια υπεγράφησαν ανά ασφαλιστική κατηγορία και ανά έτος για την πενταετία 2016-2020. Παίρνουμε από τον πίνακα *contracts* το πλήθος (count) των κωδικών συμβολαίων (ώστε να βρούμε πόσα συμβόλαια υπάρχουν στην βάση μας), την ασφαλιστική ομάδα-κατηγορία και το έτος (μέσω της εντολής *EXTRACT*) από την ημερομηνία λήξης (*end_date*). Με τη βοήθεια του *WHERE* εξετάζουμε αν τα συμβόλαια υπεγράφησαν πριν το 2016 ή μετά το 2020 καθώς και αν το έτος λήξης τους βρίσκεται ανάμεσα στο 2016 και στο 2020. Τέλος με το *GROUP BY* κατηγοριοποιούμε τα αποτελέσματα με βάση την ασφαλιστική ομάδα-κατηγορία και το έτος λήξης των συμβολαίων.

Αποτέλεσμα:

	number_of_contracts bigint		insurance_category character varying (20)		expired_year double precision
1	1	1	mixed		2017
2	2	2	private		2020
3	1	1	professional		2016
4	1	1	professional		2017
5	1	1	professional		2019
6	1	1	professional		2020

Ερώτημα 2d: Ποια κατηγορία ασφάλισης παρουσιάζει βάσει των συμβολαίων τον μεγαλύτερο τζίρο (2 παραλλαγές: σε απόλυτους αριθμούς, με αναγωγή βάσει πλήθους συμβολαίων).

- Πρώτη παραλλαγή - σε απόλυτους αριθμούς:

```
-- Ερώτημα 2d (Πρώτη παραλλαγή - σε απόλυτους αριθμούς)
WITH sum_per_category (insurance_category,sum_cost) AS
(
SELECT insurance_category,SUM(contract_cost) AS sum_cost
FROM contracts
GROUP BY insurance_category
)
SELECT insurance_category,sum_cost
FROM sum_per_category
WHERE sum_cost IN
(
SELECT MAX(sum_cost)
FROM sum_per_category
)
```


Στο συγκεκριμένο ερώτημα αναζητούμε ποια κατηγορία παρουσιάζει βάσει των συμβολαίων τον μεγαλύτερο τζίρο (με βάση το συνολικό ποσό όλων των συμβολαίων ανά κατηγορία). Με τη βοήθεια του WITH παίρνουμε από τον πίνακα *contracts* το συνολικό ποσό κάθε ασφαλιστικής ομάδας-κατηγορίας, διότι με την αθροιστική συνάρτηση SUM προσθέτουμε τα ποσά κάθε ασφαλιστικής ομάδας-κατηγορίας αφού κατηγοριοποιούμε τα αποτελέσματα βάσει αυτής. Ύστερα από το τμήμα WITH βρίσκουμε το μέγιστο συνολικό ποσό (μέσω του MAX) που αντιστοιχεί σε κάποια ασφαλιστική ομάδα-κατηγορία και εμφανίζουμε αυτό το ποσό καθώς και την ομάδα-κατηγορία.

Αποτέλεσμα:

	insurance_category character varying (20)	sum_cost bigint
1	mixed	248919776

- Δεύτερη παραλλαγή - με αναγωγή βάσει πλήθους συμβολαίων:

```
-- Ερώτημα 2d (Δεύτερη παραλλαγή - με αναγωγή βάσει πλήθους συμβολαίων)
WITH count_per_category (insurance_category,count_contracts) AS
(
  SELECT insurance_category,COUNT(DISTINCT contract_code) AS count_contracts
  FROM contracts
  GROUP BY insurance_category
)
SELECT insurance_category,count_contracts
FROM count_per_category
WHERE count_contracts IN
(
  SELECT MAX(count_contracts)
  FROM count_per_category
)
```

Στο συγκεκριμένο ερώτημα αναζητούμε ποια κατηγορία παρουσιάζει βάσει των συμβολαίων τον μεγαλύτερο τζίρο (με βάση το πλήθος συμβολαίων). Με την βοήθεια του WITH παίρνουμε από τον πίνακα *contracts* το πλήθος των συμβολαίων χρησιμοποιώντας την αθροιστική συνάρτηση COUNT. Από το τμήμα WITH που δημιουργήσαμε παίρνουμε μέσω της συνάρτησης MAX το μέγιστο πλήθος συμβολαίων από τις τρεις (3) ασφαλιστικές ομάδες-κατηγορίες, το οποίο εμφανίζεται όπως και η αντίστοιχη ομάδα-κατηγορία.

Αποτέλεσμα:

	insurance_category character varying (20)	count_contracts bigint
1	mixed	46

Ερώτημα 2ε: Ποιος είναι ο μέσος όρος συμβολαίων ανά ηλικιακή ομάδα οχημάτων (παλαιότητα 0-4 έτη, 5-9 έτη, 10-19 έτη, 20+ έτη).

```
-- Ερώτημα 2ε
-- 1η παραδοχή: Θεωρούμε ότι δεν υπάρχουν αυτοκίνητα που κυκλοφόρησαν πρώτη φορά πριν τον 1980
-- 2η παραδοχή: Όσα οχήματα υπάρχουν, τόσα είναι και τα διαφορετικά συμβόλαια
WITH contracts_0_4 (first_year,count_contracts) AS
(
    SELECT first_year, COUNT(DISTINCT plate_number) AS count_contracts
    FROM vehicles
    WHERE (SELECT date_part('year',CURRENT_DATE))-first_year <= 4
    GROUP BY first_year
),
contracts_5_9 (first_year,count_contracts) AS
(
    SELECT first_year, COUNT(DISTINCT plate_number) AS count_contracts
    FROM vehicles
    WHERE (SELECT date_part('year',CURRENT_DATE))-first_year <= 9 AND (SELECT date_part('year',CURRENT_DATE))-first_year >= 5
    GROUP BY first_year
),
contracts_10_19 (first_year,count_contracts) AS
(
    SELECT first_year, COUNT(DISTINCT plate_number) AS count_contracts
    FROM vehicles
    WHERE (SELECT date_part('year',CURRENT_DATE))-first_year <= 19 AND (SELECT date_part('year',CURRENT_DATE))-first_year >= 10
    GROUP BY first_year
),
contracts_20plus (first_year,count_contracts) AS
(
    SELECT first_year, COUNT(DISTINCT plate_number) AS count_contracts
    FROM vehicles
    WHERE (SELECT date_part('year',CURRENT_DATE))-first_year >=20
    GROUP BY first_year
)

(SELECT '0-4' AS group,CAST(SUM(count_contracts)/5 AS DECIMAL(10,2)) AS average_contracts
FROM contracts_0_4)
UNION
(SELECT '5-9' AS group,CAST(SUM(count_contracts)/5 AS DECIMAL(10,2)) AS average_contracts
FROM contracts_5_9)
UNION
(SELECT '10-19' AS group,CAST(SUM(count_contracts)/10 AS DECIMAL(10,2)) AS average_contracts
FROM contracts_10_19)
UNION
(SELECT '20+' AS group,CAST(SUM(count_contracts)/((SELECT date_part('year',CURRENT_DATE))-1999) AS DECIMAL(10,2)) AS average_contracts
FROM contracts_20plus)
```

Για την υλοποίηση του συγκεκριμένου ερωτήματος, θεωρούμε δύο (2) παραδοχές: 1^η παραδοχή: δεν υπάρχουν αυτοκίνητα που κυκλοφόρησαν πρώτη φορά πριν τον 1980 και 2^η παραδοχή: όσα οχήματα υπάρχουν, τόσα είναι και τα διαφορετικά συμβόλαια (δηλαδή ο αριθμός των διακριτών οχημάτων είναι ίσος με τον αριθμό των διακριτών συμβολαίων που υπάρχουν στην βάση).

Σε αυτό το ερώτημα αναζητούμε τον μέσο όρο συμβολαίων ανά ηλικιακή ομάδα οχημάτων. Μέσω του τμήματος WITH δημιουργούμε τέσσερα

(4) διαφορετικά αποτελέσματα ανά τις ηλικιακές ομάδες των οχημάτων. Επιλέγουμε από τον πίνακα οχήματα το έτος πρώτης κυκλοφορίας (*first_year*) και το πλήθος των αριθμών πλαισίων (*plate_number*) με τη βοήθεια της συνάρτησης COUNT. Στη συνέχεια μέσω του WHERE αφαιρούμε την τωρινή χρονιά (την οποία την παίρνουμε μέσω της εντολής SELECT date_part('year',CURRENT_DATE))) από το έτος πρώτης κυκλοφορίας και εξετάζουμε αν το αποτέλεσμα ανήκει στην αντίστοιχη ηλικιακή ομάδα (0-4, 5-9, 10-19, 20+) με το κάθε ξεχωριστό τμήμα WITH. Αφού βρούμε το άθροισμα του πλήθους των αριθμών πλαισίων μέσω της συνάρτησης SUM, τότε το διαιρούμε ανά τον αριθμό των χρόνων στην αντίστοιχη ηλικιακή ομάδα (π.χ 0-4 -> 5 χρόνια) για να υπολογίσουμε τον μέσο όρο (δεν έγινε χρήση της συναθροιστικής συνάρτησης AVG, καθώς θεωρούμε πως υπάρχει και αυτή η εναλλακτική για τον υπολογισμό του μέσου όρου). Τέλος τα κατατάσσουμε με την εντολή GROUP BY ανά έτος πρώτης κυκλοφορίας.

Αποτέλεσμα:

	group text	average_contracts numeric (10,2)
1	5-9	5.00
2	20+	0.95
3	10-19	4.10
4	0-4	2.60

Ερώτημα 2f: Ποιος είναι ο μέσος όρος συμβάντων-παραβάσεων ανά ηλικιακή ομάδα οδηγών (18- 24, 25-49, 50-69, 70+).

```

-- Ερώτημα 2f
-- Παραδοχή: Δεν υπάρχει οδηγός που να είναι άνω των 80 ετών

WITH infringements_18_24 (birth_date,infringements_number) AS
(
SELECT EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date) AS age,COUNT(infringement_code) AS infringements_number
FROM infringements NATURAL JOIN infringements_drivers NATURAL JOIN drivers_info
WHERE (EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date)) >= 18 AND (EXTRACT(YEAR FROM infringement_date)-EXTRACT (YEAR FROM birth_date)) <= 24
GROUP BY EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date)
),
infringements_25_49 (birth_date,infringements_number) AS
(
SELECT EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date) AS age,COUNT(infringement_code) AS infringements_number
FROM infringements NATURAL JOIN infringements_drivers NATURAL JOIN drivers_info
WHERE (EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date)) >= 25 AND (EXTRACT(YEAR FROM infringement_date)-EXTRACT (YEAR FROM birth_date)) <= 49
GROUP BY EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date)
),
infringements_50_69 (birth_date,infringements_number) AS
(
SELECT EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date) AS age,COUNT(infringement_code) AS infringements_number
FROM infringements NATURAL JOIN infringements_drivers NATURAL JOIN drivers_info
WHERE (EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date)) >= 50 AND (EXTRACT(YEAR FROM infringement_date)-EXTRACT (YEAR FROM birth_date)) <= 69
GROUP BY EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date)
),
infringements_70plus (birth_date,infringements_number) AS
(
SELECT EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date) AS age,COUNT(infringement_code) AS infringements_number
FROM infringements NATURAL JOIN infringements_drivers NATURAL JOIN drivers_info
WHERE (EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date)) >= 70
GROUP BY EXTRACT(YEAR FROM infringement_date)-EXTRACT(YEAR FROM birth_date)
)

(SELECT '18-24' AS age_group,CAST(SUM(infringements_number)/7 AS DECIMAL(10,2)) AS average_contracts
FROM infringements_18_24)
UNION
(SELECT '25-49' AS age_group,CAST(SUM(infringements_number)/25 AS DECIMAL(10,2)) AS average_contracts
FROM infringements_25_49)
UNION
(SELECT '50-69' AS age_group,CAST(SUM(infringements_number)/20 AS DECIMAL(10,2)) AS average_contracts
FROM infringements_50_69)
UNION
(SELECT '70+' AS age_group,CASI(SUM(infringements_number)/11 AS DECIMAL(10,2)) AS average_contracts
FROM infringements_70plus)

```

Για την υλοποίηση του συγκεκριμένου ερωτήματος, στηριζόμαστε στην ακόλουθη παραδοχή: δεν υπάρχει οδηγός (στην βάση) που να είναι άνω των 80 ετών.

Σε αυτό το ερώτημα αναζητούμε τον μέσο όρο συμβάντων-παραβάσεων ανά ηλικιακή ομάδα οδηγών (όχι πελατών-οδηγών). Μέσω του τμήματος WITH δημιουργούμε τέσσερα (4) διαφορετικά αποτελέσματα ανά τις ηλικιακές ομάδες των οδηγών. Επιλέγουμε την διαφορά του έτους που συνέβη η παράβαση (*infringement_date*) μείον το έτος που γεννήθηκε ο οδηγός (*birth_date*) χρησιμοποιώντας την εντολή EXTRACT (ώστε να βρούμε την ηλικία που είχε ο οδηγός όταν διέπραξε την παράβαση) και το πλήθος των κωδικών παραβάσεων (*infringement_code*) κάνοντας NATURAL JOIN τους πίνακες παραβάσεις (*infringements*), παραβάσεις-οδηγοί (*infringements-drivers*) και πληροφορίες οδηγών (*drivers_info*). Χρησιμοποιώντας το WHERE εξετάσουμε αν η ηλικία του οδηγού που διέπραξε την παράβαση ανήκει στην αντίστοιχη ηλικιακή ομάδα (18-24, 25-49, 50-69, 70+) με το κάθε ξεχωριστό τμήμα του WITH. Αφού βρούμε το άθροισμα του πλήθους των ηλικιών των οδηγών που διέπραξαν κάποια παράβαση μέσω της συνάρτησης SUM, τότε το διαιρούμε ανά τον αριθμό των χρόνων στην αντίστοιχη ηλικιακή ομάδα (π.χ 18-24 -> 7 χρόνια) για να υπολογίσουμε τον μέσο όρο (δεν έγινε χρήση της συναθροιστικής συνάρτησης AVG, καθώς θεωρούμε πως υπάρχει και αυτή η εναλλακτική για τον υπολογισμό του μέσου όρου). Τέλος τα κατατάσσουμε με την εντολή GROUP BY ανά την ηλικία που είχε ο οδηγός όταν διέπραξε την παράβαση.

Αποτέλεσμα:

	age_group text	average_contracts numeric (10,2)
1	50-69	2.10
2	70+	2.36
3	25-49	0.88
4	18-24	0.86

3 Υλοποίηση triggers και cursors (ερώτημα 3)

3.1 Κατασκευή trigger (ερώτημα 3α)

Για το συγκεκριμένο ερώτημα, υλοποιήσαμε έναν trigger, ο οποίος εκτελεί την ανανέωση όλων των συμβολαίων που αναμένεται να λήξουν ακριβώς στην τρέχουσα ημερομηνία. Συγκεκριμένα, όλα τα συμβόλαια υπάρχουν στον πίνακα *contracts* της βάσης. Το πεδίο *end_date* του πίνακα αυτού, αναφέρει την ημερομηνία που λήγει το συμβόλαιο. Εάν η ημερομηνία λήξης (*end_date*) ισούται με την σημερινή, τότε γίνεται ανανέωση κατά ένα (1) έτος μέσω του trigger.

Παρακάτω, φαίνεται η συνάρτηση που εκτελεί το trigger:

```
-- Η συνάρτηση που θα εκτελεί το trigger
CREATE OR REPLACE FUNCTION renew_contract_year() RETURNS TRIGGER AS
$$
BEGIN
    -- Έλεγχος για το αν η κατηγορία του συμβολαίου είναι 'professional'
    IF OLD.insurance_category='professional' THEN
        -- Ανανέωση του συμβολαίου κατά ένα χρόνο
        NEW.end_date=CAST(NEW.end_date +interval '1 year' AS date);
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;
```

Ουσιαστικά, εάν η κατηγορία του συμβολαίου (*insurance_category*) είναι “επαγγελματική” (professional), τότε στην ημερομηνία λήξης του συμβολαίου προστίθεται ένας επιπλέον χρόνος.

Όσο αφορά τον ορισμό του trigger, ενδεικτικά έχουμε την ακόλουθη δήλωση:

```
-- Trigger που θα εκτελεί την ανανέωση όλων των συμβολαίων στην κατηγορία
-- 'επαγγελματικό' κατά έναν χρόνο. Η ανανέωση αυτή γίνεται μόνο όταν
-- κάνουμε UPDATE στον πίνακα contracts
CREATE TRIGGER renew
BEFORE UPDATE
ON contracts
FOR EACH ROW
-- Ο trigger πυροδοτείται μόνο όταν η ημερομηνία λήξης του συμβολαίου
-- είναι η σημερινή (τρέχουσα)
WHEN (old.end_date = CURRENT_DATE)
EXECUTE PROCEDURE renew_contract_year();
```

Το CURRENT_DATE ουσιαστικά μεταφράζεται ως την τρέχουσα ημερομηνία (22/06/2021 όταν γράφηκε η τεκμηρίωση). Στην συνέχεια, παρουσιάζεται ένα παράδειγμα υλοποίησης του trigger:

	contract_code [PK] character varying (20)	insurance_category character varying (20)	start_date date	end_date date	contract_cost integer
1	A087805	professional	2018-06-22	2021-06-22	42008

Εγγραφή πριν από την πυροδότηση του trigger

Στην παραπάνω εικόνα, έχουμε μία εγγραφή του πίνακα συμβόλαια. Πρόκειται για ένα συμβόλαιο της κατηγορίας “professional”, το οποίο αναμένεται να λήξει στις 22/06/2021 (τρέχουσα ημερομηνία). Μετά την εκτέλεση του ακόλουθου UPDATE query, ενεργοποιείται ο trigger renew (έχει δηλωθεί ως BEFORE UPDATE):

```
-- Χρησιμοποιούμε ένα UPDATE query για να δείξουμε ότι ενεργοποιείται το trigger
UPDATE contracts SET contract_cost='42008' where contract_code='A087805'
```

Το αποτέλεσμα παρουσιάζεται στην επόμενη εικόνα. Το συμβόλαιο ανανεώθηκε κατά έναν χρόνο:

	contract_code [PK] character varying (20)	insurance_category character varying (20)	start_date date	end_date date	contract_cost integer
1	A087805	professional	2018-06-22	2022-06-22	420010

Η ίδια εγγραφή μετά από την πυροδότηση του trigger

3.2 Κατασκευή cursor (ερώτημα 3b)

Για το συγκεκριμένο ερώτημα, υλοποιήσαμε με την βοήθεια cursors το υποερώτημα 2c (όχι την παραλλαγή). Για τις ανάγκες αυτού του ζητήματος, ορίστηκε μία συνάρτηση `get_contracts(year1 int, year2 int)`, η οποία επιστρέφει έναν πίνακα ως αποτέλεσμα. Επιπρόσθετα, έχει δηλωθεί ένας cursor με όνομα `cur_contracts`, ο οποίος σχετίζεται με το ερώτημα 2c. Τα ορίσματα `year1` και `year2` της συνάρτησης περνάνε ουσιαστικά σε αυτό το ερώτημα του cursor.

Κατόπιν, η συνάρτηση κατασκευάζει τον τελικό πίνακα, τον οποίο και επιστρέφει. Ο πίνακας αυτός περιέχει τα αποτελέσματα του ερωτήματος 2c. Φυσικά, το αποτέλεσμα εξαρτάται και από τα ορίσματα που δίδει κάθε φορά ο χρήστης στην συνάρτηση.

Παρακάτω, φαίνεται η εκτέλεση αυτής της συνάρτησης με τη βοήθεια του ακόλουθου ερωτήματος:

```
-- Ενεργοποίηση της παραπάνω συνάρτησης με ορίσματα 2016 και 2020
SELECT * FROM get_contracts(2016,2020);
```

Κλήση της συνάρτησης με ορίσματα 2016 και 2020

	nr_of_contracts bigint	insurance_ctgr character varying	signed_in integer
1	2	mixed	2016
2	4	mixed	2017
3	2	mixed	2018
4	1	mixed	2019
5	1	mixed	2020
6	1	private	2016
7	1	private	2017
8	1	private	2019
9	2	professional	2016
10	2	professional	2017
11	3	professional	2018
12	2	professional	2019
13	1	professional	2020

Αποτελέσματα της συνάρτησης

4 Σύνδεση βάσης δεδομένων με API (ερώτημα 4)

Για το συγκεκριμένο ερώτημα, έχουμε κάνει την σύνδεση της βάσης δεδομένων, με την γλώσσα pythοn (μέσω της βιβλιοθήκης psycopg2). Αρχικά, το πρόγραμμα εκτελεί την σύνδεση με την βάση δεδομένων Insurance_Company. Τα στοιχεία της σύνδεσης είναι τα ακόλουθα:

- Username: postgres
- Hostname: 127.0.0.1
- Port number: 5432
- Database name: Insurance_Company

Ωστόσο, το πρόγραμμα επιτρέπει στον χρήστη να εισάγει τις δικές του τιμές για τα ανωτέρω σε περίπτωση διαφορετικών στοιχείων σύνδεσης.

Στην συνέχεια, εκτελούνται όλα τα υποερωτήματα του ερωτήματος 2 (συνολικά είναι οκτώ (8) μαζί με τις παραλλαγές) ένα προς ένα. Τα

υποερωτήματα εκτελούνται με την βοήθεια ενός cursor, τον οποίο έχουμε δεσμεύσει στην αρχή του προγράμματος. Όταν όλα τα queries εκτελεστούν με επιτυχία, ο cursor αποδεσμεύεται και το πρόγραμμα τερματίζεται.

Σε περίπτωση αποτυχίας εκτέλεσης (λόγω προβλημάτων σύνδεσης ή αποτυχία εκτέλεσης των ερωτημάτων) το πρόγραμμα επιστρέφει exception στον χρήστη και έπειτα τερματίζεται.

Παρακάτω φαίνεται η εκτέλεση του προγράμματος μέσω του CMD Windows:

```
PostgreSQL successfully connected with python API!
2a:
first_name = Patricia , last_name = Orcutt , contract_code = A484319 , category = Driver
first_name = Manuel , last_name = Ladaille , contract_code = A484319 , category = Client
first_name = Erik , last_name = Chapman , contract_code = A484319 , category = Driver
first_name = Jose , last_name = Wood , contract_code = A647907 , category = Driver
first_name = Ruth , last_name = Zimmerman , contract_code = A647907 , category = Client
first_name = Charlie , last_name = Nelson , contract_code = A975011 , category = Client
first_name = Giel , last_name = Watson , contract_code = A975011 , category = Driver
first_name = Daan , last_name = Pensec , contract_code = A975011 , category = Driver
first_name = Marcin , last_name = Knight , contract_code = P209751 , category = Client
first_name = Eleanor , last_name = Linhart , contract_code = P209751 , category = Driver
first_name = Megan , last_name = Pierce , contract_code = P270842 , category = Driver
first_name = Marta , last_name = Wakefield , contract_code = P270842 , category = Client
first_name = Marty , last_name = Poole , contract_code = P270842 , category = Driver
first_name = Teun , last_name = Gibson , contract_code = P329398 , category = Driver
first_name = Erik , last_name = Chapman , contract_code = P329398 , category = Driver
first_name = Megan , last_name = Hankins , contract_code = P329398 , category = Client
first_name = Megan , last_name = Pierce , contract_code = P329398 , category = Driver
first_name = Sally , last_name = DelRosso , contract_code = P474722 , category = Client
first_name = Al , last_name = Guyer , contract_code = P474722 , category = Driver
first_name = Peg , last_name = Petrzalka , contract_code = P590699 , category = Driver
first_name = Bob , last_name = King , contract_code = P590699 , category = Client
first_name = Duncan , last_name = Weaver , contract_code = P671261 , category = Client
first_name = Ed , last_name = Weaver , contract_code = P671261 , category = Driver
first_name = Shermie , last_name = Wilson , contract_code = P671261 , category = Driver
first_name = Sylvia , last_name = Gibson , contract_code = P774736 , category = Driver
first_name = Jonas , last_name = Overton , contract_code = P774736 , category = Client
first_name = Edwina , last_name = Stewart , contract_code = P811636 , category = Driver
first_name = Stanislaw , last_name = Thompson , contract_code = P811636 , category = Client
```

```
2b:
contract_code = A647907 , phone_number1 = 204-3388 , phone_number2 = 621-5513
contract_code = A925510 , phone_number1 = 807-971-2943 , phone_number2 = 1-252-887-9180
contract_code = A298941 , phone_number1 = 1-416-443-3339 , phone_number2 = 1-551-5610
contract_code = A701776 , phone_number1 = 1-322-270-2720 , phone_number2 = 865-6349
contract_code = P914992 , phone_number1 = 1-030-971-1181 , phone_number2 = 1-642-9174
contract_code = A625101 , phone_number1 = 1-689-5740 , phone_number2 = None
```

```
-----

2c:
number_of_contracts = 2 , insurance_category = mixed , contract_year = 2016.0
number_of_contracts = 4 , insurance_category = mixed , contract_year = 2017.0
number_of_contracts = 2 , insurance_category = mixed , contract_year = 2018.0
number_of_contracts = 1 , insurance_category = mixed , contract_year = 2019.0
number_of_contracts = 1 , insurance_category = mixed , contract_year = 2020.0
number_of_contracts = 1 , insurance_category = private , contract_year = 2016.0
number_of_contracts = 1 , insurance_category = private , contract_year = 2017.0
number_of_contracts = 1 , insurance_category = private , contract_year = 2019.0
number_of_contracts = 2 , insurance_category = professional , contract_year = 2016.0
number_of_contracts = 2 , insurance_category = professional , contract_year = 2017.0
number_of_contracts = 3 , insurance_category = professional , contract_year = 2018.0
number_of_contracts = 2 , insurance_category = professional , contract_year = 2019.0
number_of_contracts = 1 , insurance_category = professional , contract_year = 2020.0
```

```
2c-variation:
number_of_contracts = 1 , insurance_category = mixed , expired_year = 2017.0
number_of_contracts = 2 , insurance_category = private , expired_year = 2020.0
number_of_contracts = 1 , insurance_category = professional , expired_year = 2016.0
number_of_contracts = 1 , insurance_category = professional , expired_year = 2017.0
number_of_contracts = 1 , insurance_category = professional , expired_year = 2019.0
number_of_contracts = 1 , insurance_category = professional , expired_year = 2020.0
```

```
-----

2d-first variation:
insurance_category = mixed , sum_cost = 248919776
```

```
-----

2d-second variation:
insurance_category = mixed , count_contracts = 46
```

```
-----

2e:
group = 5-9 , average_contracts = 5.00
group = 20+ , average_contracts = 0.95
group = 10-19 , average_contracts = 4.10
group = 0-4 , average_contracts = 2.60
```

```
2f:
age_group = 50-69 , average_contracts = 2.10
age_group = 70+ , average_contracts = 2.36
age_group = 25-49 , average_contracts = 0.88
age_group = 18-24 , average_contracts = 0.86
```

```
PostgreSQL connection is closed
Press ENTER to EXIT_
```