Exercise of Function

Exercise 1: Find Characters

Provider: Stephane Faroult

Description:

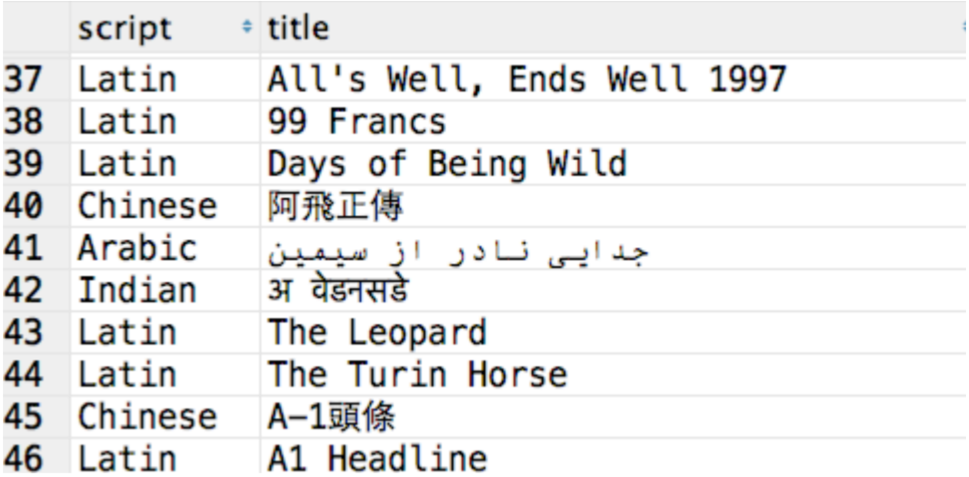
Writing a function that recognizes in which script a text is written (to be applied to column TITLE in table ALT_TITLES). We'll only consider the main writing systems. Here is a reference:

List of writing systems - Wikipedia

In particular the table at List of writing scripts by adoption, with the number of users. If you execute the query:

```
select script (title ),title from (select title from alt_titles) x
```

The result would be



	script	title
37	Latin	All's Well, Ends Well 1997
38	Latin	99 Francs
39	Latin	Days of Being Wild
40	Chinese	阿飛正傳
41	Arabic	جدایی نادر از سیمین
42	Indian	अ वेडनसडे
43	Latin	The Leopard
44	Latin	The Turin Horse
45	Chinese	A–1頭條
46	Latin	A1 Headline

Hints: Ranges to consider for the ascii()

return value (approximate blocks but the result should be OK - can be refined if needed)

Latin <=740 [7424,8594] [11360,11391] [42786,43876]

Greek [880, 1023] [7462, 8446]

Cyrillic [1024, 1327] [7296, 7544] [42560, 42655]

Arabic [1536, 2303] [64336, 69246] [124464, 126705]

Indian [2304, 3572]

Thai [3585, 3675]

Burmese [4096, 4255]

Korean [4352, 4607] [12593, 12686] [12800, 12926] [43360, 55203] [43360, 55291] [65440, 65500]

Khmer [6016, 6137]

Chinese [11904, 12333] [12344, 12347] [13312, 42182]

Japanese [12353, 12543] [12784, 12799] [13008, 13143]

Other everything else ...

Sample solution:

```
create function script(fm character varying)
 returns character varying
language plpgsql
as $$
declare
 ascimax int;
 ascimin int;
  value varchar;
begin
 select max(x.a)
 into ascimax
 from (
 select distinct
 ascii(substr(t.title, n, 1)) a,
 substr(t.title, n, 1)
 from (select fm as title) t
 cross join generate_series(1, length(t.title)) n) x;
 select min(x.a)
 into ascimin
 from (
 select distinct
 ascii(substr(t.title, n, 1)) a,
 substr(t.title, n, 1)
 from (select fm as title) t
 cross join generate_series(1, length(t.title)) n) x
 where a > 127;
 if ((ascimax <= 740 and ascimax >= 0) or (ascimax >= 7424 and ascimax <=
8594) or
 (ascimax >= 11360 and ascimax <= 11391) or (ascimax >= 42786 and ascimax
<= 43876))
 then value = 'Latin';
 elseif ((ascimin >= 880 and ascimin <= 1023) or (ascimin >= 7462 and ascimin
<= 8446))
 then value = 'Greek';
 elseif ((ascimin >= 1024 and ascimin <= 1327) or (ascimin >= 7296 and ascimin
<= 7544) or
 (ascimin >= 42560 and ascimin <= 42655))
 then value = 'Cyrillic';
 elseif ((ascimin >= 1536 and ascimin <= 2303) or (ascimin >= 64336 and
ascimin <= 69246) or
 (ascimin >= 124464 and ascimin <= 126705))
 then value = 'Arabic';
 elseif (ascimin >= 2304 and ascimin <= 3572)
 then value = 'Indian';
 elseif (ascimin >= 3585 and ascimin <= 3675)
 then value = 'Thai';
 elseif (ascimin >= 4096 and ascimin <= 4255)
 then value = 'Burmese';
 elseif ((ascimin >= 4352 and ascimin <= 4607) or (ascimin >= 12593 and
ascimin <= 12686) or
 (ascimin >= 12800 and ascimin <= 12926) or (ascimin >= 43360 and
ascimin <= 55203) or
 (ascimin >= 43360 and ascimin <= 55291) or (ascimin >= 65440 and
ascimin <= 65500))
 then value = 'Korean';
 elseif (ascimin >= 6016 and ascimin <= 6137)
 then value = 'Khmer';
 elseif ((ascimin >= 11904 and ascimin <= 12333) or (ascimin >= 12344 and
ascimin <= 12347) or
 (ascimin >= 13312 and ascimin <= 42182))
 then value = 'Chinese';
 elseif ((ascimin >= 12353 and ascimin <= 12543) or (ascimin >= 12784 and
ascimin <= 12799) or
 (ascimin >= 13008 and ascimin <= 13143))
 then value = 'Japanese';
 else value = 'Other';
 end if;
 return value;
end;
$$;
```

Exercise 2: Car Number:

Providers: ZHU Yueming, ZHAO Yi

Description:

You need to finish one of following two task.

Task 1. submit any screenshot of learning this week.

Task 2. finish the question below and submit a sql file

Design a function named `get_city(car_num varchar)` that can return the city according to the license plate number of a car.

In your function, you should verify that the license plate number started by a valid license Vehicle registration institution code (for example: '粤A') and then have **5** or **6** characters (for example: '88888' or 'D88888')

In this exercise, we only need to consider following seven cities in Guang Dong province, which means for other input of registraction institution code can be regarded as invalid input.

city	code
GUANG ZHOU	粤A
SHEN ZHEN	粤B
ZHU HAI	粤C
SHAN TOU	粤D
FO SHAN	粤E
SHAO GUAN	粤F
ZHAN JIANG	粤G

If there are 5 characters after the license Vehicle registration institution code, all the characters should be either capital English letters or numbers in the range of 0~9, inclusive.

If there are 6 characters after the license Vehicle registration institution code, the first character should be either 'D' or 'F', and other characters should be either capital English letters or numbers in the range of 0~9, inclusive.

If the license plate number is valid return the corresponding city, otherwise throw exceptions "Invalid Province", "Invalid car number!","Invalid plate length","Invalid city" respectively.

Sample input and output

parameter	exception or result
京A11111	Invalid Province
粤A111111	Invalid plate Length
粤A1111	Invalid plate Length
粤H1111	Invalid plate Length
粤GF11111	ZHAN JIANG
粤AD11111	GUANG ZHOU
粤AR11111	Invalid plate Length
粤ARRRRR	GUANG ZHOU
粤Arrrrr	Invalid car number

Sample solutions:

Sample Solution 1:

```
create function get_city(car_num varchar)
    returns varchar
    language plpgsql
as
$$
DECLARE
    cn_len      INT;
    i           INT         := 3;
    city_number INT;
    ascii       INT;
    city_names  varchar(80) := 'GUANG ZHOU,SHEN ZHEN,ZHU HAI,SHAN TOU,FO
SHAN,SHAO GUAN,ZHAN JIANG';
BEGIN
    -- check length
    if (substring(car_num, 1, 1) <> '粤')
    then
        raise exception 'Invalid Province';
    end if;

    cn_len = char_length(car_num);
    if not
        cn_len between 7 and 8
    then
        raise exception 'Invalid plate Length';
    elseif
        cn_len = 8 and substring(car_num, 3, 1) not in ('F', 'D')
    then
        raise exception 'Invalid plate Length';
    end if;
    -- check car number
    while i <= cn_len
        LOOP
            ascii = ascii(substring(car_num, i, 1));
            if not (ascii >= 48 and ascii <= 57 or ascii <= 90 and ascii >= 65)
            then
                raise exception 'Invalid car number';
            END IF;
            i = i + 1;
        END LOOP;
    -- check city info
    city_number = ascii(substring(car_num, 2, 1));
    if city_number not between 65 and 71 then
        raise exception 'Invalid city';
    else
        return split_part(city_names, ',', city_number - 64);
    end if;
END;
$$;
```

Sample Solution2:

```
create function get_city(car_num varchar)
    returns varchar
    language plpgsql
as
$$
DECLARE
    cn_len      INT;
    city_number INT;
    city_names  varchar(80) := 'GUANG ZHOU,SHEN ZHEN,ZHU HAI,SHAN TOU,FO
SHAN,SHAO GUAN,ZHAN JIANG';
BEGIN
```

```
    -- check length
    if (substring(car_num, 1, 1) <> '粤')
    then
        raise exception 'Invalid Province';
    end if;

    cn_len = char_length(car_num);
    if not
        cn_len between 7 and 8
    then
        raise exception 'Invalid plate Length';
    elseif
        cn_len = 8 and substring(car_num, 3, 1) not in ('F', 'D')
    then
        raise exception 'Invalid plate Length';
    end if;
    -- check car number
    if not substring(car_num, cn_len - 4, 5) ~ '^[A-Z0-9]+$'
    then
        raise exception 'Invalid car number';
    end if;
    city_number = ascii(substring(car_num, 2, 1));
    if city_number not between 65 and 71 then
        raise exception 'Invalid city';
    else
        return split_part(city_names, ',', city_number - 64);
    end if;
END;
$$;
```