Oracle Database - programming
Task list No. 0

Task 0. In a few sentences, an outline of a reality fragment is presented:

After many years of independence, cats of both sexes hunted in the Wólka Mała village decided to organize themselves. So, a herd was created, commanded by the most excellent mouse hunter with the nickname Tiger. As part of the herd, under the leadership of the Tiger, an informal hierarchy of cats formed naturally. Each cat knew which other cat it was subordinated to. The pack was additionally administratively divided into several, having a unique number and name, of the bands, each commanded by an outstanding mouse hunter nominated by the Tiger. Each band was given an independent area where their cats could organize their hunts. Due to their high position, the Tiger and crew members had the privilege of hunting in the entire area controlled by the herd. For identification purposes, each cat was required to choose a unique nickname. The cat should also have a name. It was agreed that a member of the herd would be rewarded with a ration of mice every month for his contribution to maintaining the entire pack. This ratio will be adequate for the function performed in the feline community. This function will define the lower and upper limits of the mice ratio. Regardless of the mice ration size, the herd leader, for special merits, will be able to grant the cat, at his own discretion, additional mice ration. Cats hunted happily in their assigned areas, but occasionally, there were incidents with representatives of other breeds. The participants of the incidents, identified by their names, automatically became the personal enemies of the wronged cats. Their degree of hostility and their species were carefully noted. All these events were described as a warning to cats and infamy for enemies (mandatory with their date). Assuming, however, that a real hunter can avoid known enemies, only the first cat incident with a specific enemy was recorded. Over time, cats noticed that handling bribes to enemies can reduce their alertness. For this reason, the sort of bribe preferred by each enemy was noted.

As a result of the analysis of this fragment or reality, the following database schema was created, consisting of five relations (**Bands**, **Functions**, **Cats**, **Enemies**, **Incidents)**:

Bands:

Band_no	NUMBER(2)	primary key
name	VARCHAR2(20)	mandatory attribute
site	VARCHAR2(15)	unique attribute
band_chief	VARCHAR2(15)	unique foreign key from Cats relation (attribute nickname)

Functions:

function	VARCHAR2(10)	primary key
min_mice	NUMBER(3)	value > 5
max_mice	NUMBER(3)	200>value>=min_mouse

Enemies:

enemy_name	VARCHAR2(15)	primary key
hostility_degree	NUMBER(2)	value from 1 to 10
species	VARCHAR2(15)	
bride	VARCHAR2(20)	

Cats:

name	VARCHAR2(15)	mandatory attribute
gender	VARCHAR2(1)	two values: 'M' and 'W'
nickname	VARCHAR2(15)	primary key
function	VARCHAR2(10)	foreign key from Function relation (attribute function)
chief	VARCHAR2(15)	foreign key from the Cats relation (attribute nickname)
in_herd_since	DATE	default value: current date value (SYSDATE)
mice_ration	NUMBER(3)	
mice_extra	NUMBER(3)	
band_no	NUMBER(2)	foreign key from Bands relation (attribute band_no)

Incidents:

nickname	VARCHAR2(15)	partial primary key, foreign key from Cats relation (attribute nickname)
enemy_name	VARCHAR2(15)	partial primary key, foreign key from Enemies relation (attribute enemy_name)
incident_date	DATE	mandatory attribute
incident_desc	VARCHAR2(50)	

Please write a script that creates the above relations in the database. The lack of information about the attribute to be mandatory means the default nonmandatory value. Please define all potential column constraints as column ones. The definitions of all constraints should be a component of the CREATE TABLE command (do not use the ALTER TABLE command unless this will be the only solution to the problem). Run the created script in the SQL Developer application. Then, create a script that fills the relations using the below data and run it in the SQL Developer application.

Data for the relation Cats(name, gender, nickname, function, chief, in_herd_since, mice_ration, mice_extra, band_no)

```
'JACEK','M','CAKE','CATCHING','BALD','2008-12-01',67,NULL,2
'BARI','M','TUBE','CATCHER','BALD','2009-09-01',56,NULL,2
'MICKA','D','LOLA','NICE','TIGER','2009-10-14',25,47,1
'LUCEK','M','ZERO','CAT','HEN','2010-03-01',43,NULL,3
'SONIA','D','FLUFFY','NICE','ZOMBIES','2010-11-18',20,35,3
'LATKA','D','EAR','CAT','REEF','2011-01-01',40,NULL,4
'DUDEK','M','SMALL','CAT','REEF','2011-05-15',40,NULL,4
'MRUCZEK','M','TIGER','BOSS',NULL,'2002-01-01',103,33,1
'CHYTRY','M','BOLEK','DIVISIVE','TIGER','2002-05-05',50,NULL,1
'KOREK','M','ZOMBIES','THUG','TIGER','2004-03-16',75,13,3
'BOLEK','M','BALD','THUG','TIGER','2006-08-15',72,21,2
'ZUZIA','D','FAST','CATCHING','BALD','2006-07-21',65,NULL,2
'RUDA','D','LITTLE','NICE','TIGER','2006-09-17',22,42,1
'PUCEK','M','REEF','CATCHING','TIGER','2006-10-15',65,NULL,4
'PUNIA','D','HEN','CATCHING','ZOMBIES','2008-01-01',61,NULL,3
'BELA','D','MISS','NICE','BALD','2008-02-01',24,28,2
'KSAWERY','M','MAN','CATCHER','REEF','2008-07-12',51,NULL,4
'MELA','D','LADY','CATCHER','REEF','2008-11-01',51,NULL,4
```

Data for the relation Bands(band_no, name, site, band_chief)

```
1,'SUPERIORS','WHOLE AREA','TIGER'
2,'BLACK KNIGHTS','FIELD','BALD'
3,'WHITE HUNTERS','ORCHARD','ZOMBIES'
4,'PINTO HUNTERS','HILLOCK','REEF'
5,'ROCKERS','FARM',NULL
```

Data for the relation Functions (function, min_mice, max_mice)

```
'BOSS',90,110
'THUG',70,90
'CATCHING',60,70
'CATCHER',50,60
'CAT',40,50
'NICE',20,30
'DIVISIVE',45,55
'HONORARY',6,25
```

Data for the relation Enemies(enemy_name, hostility_degree, species, bribe)

```
'KAZIO',10,'MAN','BOTTLE'
'STUPID SOPHIA',1,'MAN','BEAD'
'UNRULY DYZIO',7,'MAN','CHEWING GUM'
'DUN',4,'DOG','BONE'
'WILD BILL',10,'DOG',NULL
'REKS',2,'DOG','BONE'
'BETHOVEN',1,'DOG','PEDIGRIPALL'
'SLYBOOTS',5,'FOX','CHICKEN'
'SLIM',1,'PINE',NULL
'BASIL',3,'ROOSTER','HEN TO THE HERD'
```

Data for the relation Incidents(nickname, enemy_name, incident_date, incident_desc)

```
'TIGER','KAZIO','2004-10-13','HE HAS TRYING TO STICK ON THE FORK'
'ZOMBIES', 'UNRULY DYZIO','2005-03-07','HE FOOTED AN EYE FROM PROCAST'
'BOLEK','KAZIO','2005-03-29','HE CLEANED DOG'
'FAST', 'STUPID SOPHIA' ,'2006-09-12','SHE USED THE CAT AS A CLOTH'
'LITTLE','SLYBOOTS','2007-03-07','HE RECOMMENDED HIMSELF AS A HUSBAND'
'TIGER','WILD BILL','2007-06-12','HE TRIED TO KILL'
'BOLEK','WILD BILL','2007-11-10','HE BITE THE EAR'
'MISS','WILD BILL','2008-12-12','HE BITCHED'
'MISS','KAZIO','2009-01-07','HE CAUGHT THE TAIL AND MADE A WIND'
'LADY','KAZIO','2009-02-07','HE WANTED TO SKIN OFF'
'MAN','REKS','2009-04-14','HE BARKED EXTREMELY RUDELY'
'BALD','BETHOVEN','2009-05-11','HE DID NOT SHARE THE PORRIDGE'
'TUBE','WILD BILL','2009-09-03','HE TOOK THE TAIL'
'CAKE','BASIL','2010-07-12','HE PREVENTED THE CHICKEN FROM BEING HUNTED'
'FLUFFY','SLIM','2010-11-19','SHE THREW CONES'
'HEN','DUN','2010-12-14','HE CHASED'
'SMALL','SLYBOOTS','2011-07-13','HE TOOK THE STOLEN EGGS'
'EAR', 'UNRULY DYZIO','2011-07-14','HE THREW STONES'
```

The database describing the population of cats will be the basis of all tasks on the lists and also the basis of examples in the lecture.

The INSERT command for filling relations with data will be presented in the lecture. However, the fragment considering the syntax of this command is given below.

The INSERT command inserts one or more rows directly or indirectly into an existing relation. The latter occurs when insertion occurs through a simple view, also known as a modifiable perspective (both concepts will be presented later in the lecture). The syntax for the INSERT command is as follows:

INSERT INTO RelationViewName [({attribute [, …]})]
{**VALUES** ({value [, …]})} | subquery

The list of attributes specified after the comma specifies the names of the attributes that will get value. All attributes not listed must be optional (NULL) or have a default value defined (specified in the CREATE TABLE command - DDL component of SQL language). The lack of a list of attributes in the command indicates that all attributes of the relation will be filled in the order of their definition in the CREATE TABLE command. Data can be specified explicitly in the VALUES clause through a list of values specified after the comma or implicitly through the subquery. In the first case, one row is inserted into the relation. In the second case, as many rows as the rows returned by the subquery. The number of explicitly entered values and the number of values returned by the subquery must be equal to the number of attributes specified in the attribute list (if any), and the types of these values must match the types of the respective attributes.

A version of the INSERT command allows you to insert multiple rows as part of one such command. The short version of this command has the following syntax:

INSERT ALL {INTO RelationViewName [({attribute [, …]})]
VALUES ({value [, …]}) [...]}
{SELECT * FROM Dual }| subquery

The above version of the INSERT command reduces the time to load data to the database (only one connection). It can be used to batch rewrite data from one database to another when it is certain that the source data is correct. In this version, entering rows into many different relations with one command is also possible. The subquery returning rows to insert can also be a data source. In this case, the values in the VALUES clause will be the names of the expressions (their aliases) or the names of the attributes returned by the subquery.