

## Esercizio 1

Si consideri il seguente schema relazionale, relativo alla base di dati di un'agenzia di assicurazioni:

ASSICURATI(CodA, Nome, Cognome, Indirizzo, Telefono)

POLIZZE(CodP, Data, Tipo, CodA, CodM)

INCIDENTI(CodI, Data, Descrizione, Causato, CodM)

MEZZI(CodM, Tipo, Targa, AnnoI, CodA)

INFO\_POLIZZA(CodP, CodInfo, Valore)

- Il campo Causato della tabella incidenti e' un booleano che indica se l'incidente e' stato causato o subito dall'assicurato.
- Il campo AnnoI di Mezzi rappresenta l'anno di immatricolazione del mezzo.
- La tabella INFO\_POLIZZA mantiene informazioni circa condizioni contrattuali. Per ciascuna, CodInfo rappresenta il tipo di clausola (per esempio Incendio\_e\_furto) e Valore l'importo assicurato per quella clausola.

Si richiede di:

1. (punti 3) Progettare uno schema concettuale ad oggetti equivalente allo schema relazionale precedente, motivando le scelte effettuate. Si assume che le tabelle non abbiano chiavi secondarie. Lo schema deve contenere almeno un'aggregazione complessa (di tipo SET).
2. (punti 3) Tradurre lo schema concettuale in uno schema logico relazionale ad oggetti per Oracle, motivando le scelte effettuate.
3. (punti 2) Presentare gli statement di creazione della parte di schema logico relativa all'entita' concettuale POLIZZE utilizzando il sistema dei tipi di Oracle (si presentino solo le definizioni dei tipi necessari alla definizione della tabella POLIZZE ma non dei loro sottotipi).
4. (punti 3) Supponendo che le tabelle contengano un certo numero di tuple, presentare gli statement per inserire nella base di dati una nuova polizza di codice 123, stipulata in data 21/4/04, di tipo "assicurazione auto", per l'assicurato con codice 34 e mezzo con codice 42. La polizza prevede la clausola incendio e furto per un valore di 3.000 euro. La responsabilita' civile copre fino ad un valore massimale di 1.000.000 euro.
5. Presentare gli statement SQL per le seguenti interrogazioni:
  - o (punti 3) Determinare il nome, il cognome e il numero di telefono dell'assicurato che ha realizzato il maggior numero di incidenti nel 2003.
  - o (punti 3) Determinare il codice degli incidenti coperti da piu' di una polizza (questo avviene quando l'incidente si riferisce ad un assicurato che ha stipulato piu' di una polizza per lo stesso mezzo).

## Tipi in Oracle

```
CREATE TYPE Nome AS OBJECT  
( Dichiarazioni_attributi)  
[FINAL | NOT FINAL];
```

```
CREATE TYPE Nome UNDER Nome_superclasse  
( Dichiarazioni_attributi)  
[FINAL | NOT FINAL];
```

Default: NOT FINAL

```
CREATE TYPE Nome AS TABLE OF Tipo;
```

```
CREATE TYPE Nome AS VARRAY(Numero) OF Tipo;
```

```
CREATE TABLE Nome OF Nome_Tipo  
[({Vincoli chiave primaria, chiave esterna, not null, su campi nel tipo |  
SCOPE FOR NomeCampoREF is NomeTabella})]  
[NESTED TABLE NomeCampoNested STORE AS NomeTabella];
```

```
CREATE TABLE Nome  
(Dichiarazioni_attributi)  
[NESTED TABLE NomeCampoNested STORE AS NomeTabellaStorage];
```

con

- NomeCampoNested: nome del campo della tabella di tipo TABLE.
- NomeTabellaStorage: nome tabella utilizzata per la memorizzazione della tabella nested.
- I tipi utilizzati in Dichiarazioni\_attributi possono essere:
  - Tipi atomici
  - Tipi definiti dall'utente
  - REF Nome [SCOPE IS Nome\_tabella]  
La clausola SCOPE IS puo' essere inserita solo negli statement di CREATE TABLE.  
La clausola puo' anche essere aggiunta ai campi delle tabelle tipate (SCOPE FOR).

## Funzioni OR in Oracle

- ref(): dato un oggetto di un certo tipo, restituisce l'identificatore per quell'oggetto
- deref(): dato un identificatore, restituisce l'oggetto puntato
- value(): prende un alias di relazione e restituisce l'oggetto tupla associato (utilizzando il costruttore opportuno)
- TABLE(): data una valori di tipo collezione, restituisce la tabella corrispondente al contenuto di tale valore.

## Trigger SQL-99

```
CREATE TRIGGER Nome  
{BEFORE|AFTER} Evento ON Relazione  
[REFERENCING {OLD AS Var | NEW AS Var | OLD AS Var | NEW AS Var}]  
[FOR EACH {ROW | STATEMENT}]  
[WHEN Condizione ]  
Comandi SQL  
con
```

- Evento: INSERT, DELETE, UPDATE [OF Lista Attributi]
- Condizione: clausola WHERE SQL
- Se l'azione comprende piu' di un comando: BEGIN ATOMIC ... END;