Database Implementation for the BAM projects: a Primer Guide

A. Benabdelkader

University of Amsterdam

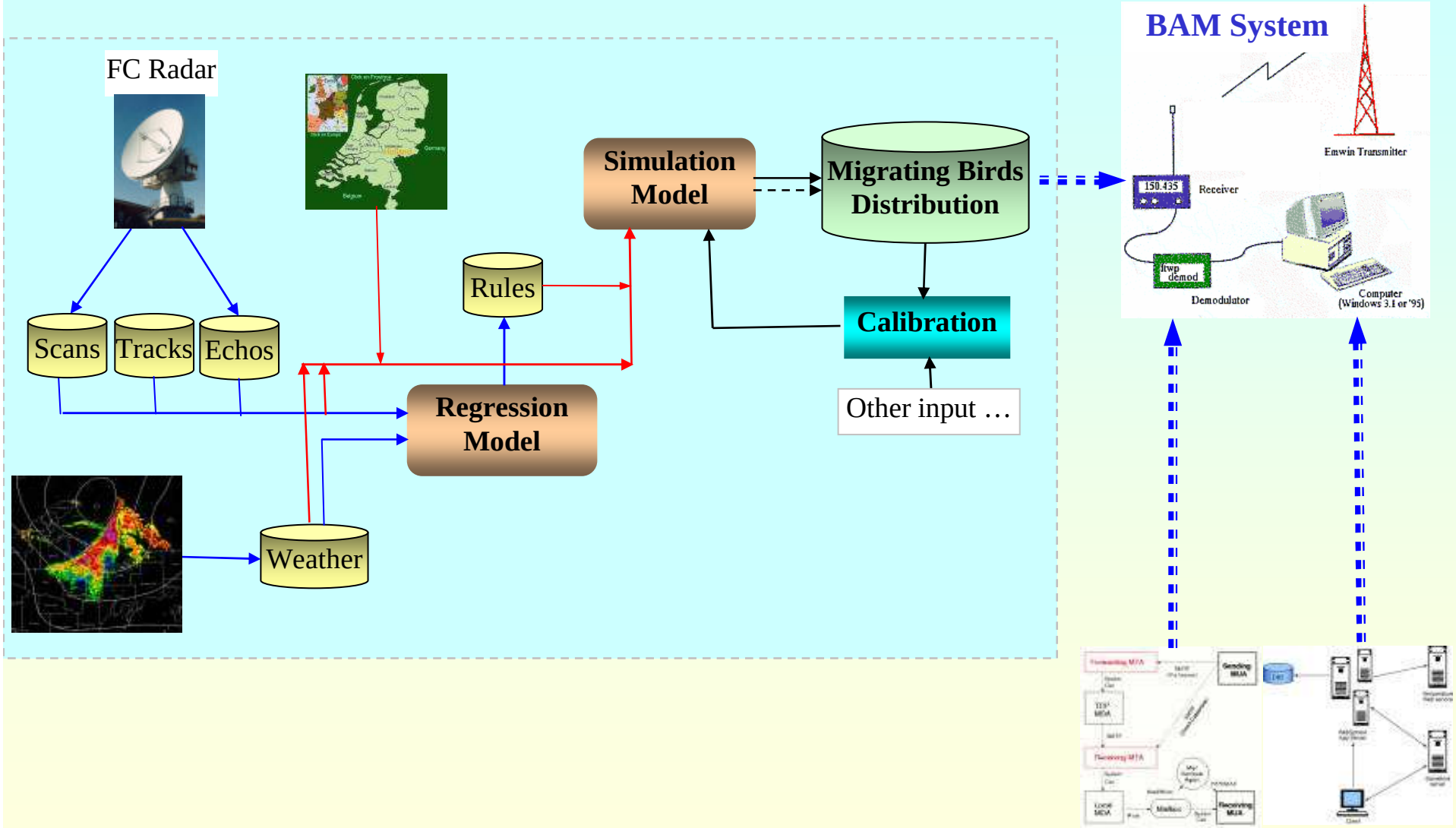
September 17, 2003

Amsterdam, NL

Presentation Outline

- An Overview of the BAM application environment
- Data Modeling: Approach
- Database Design and Implementation
  - Data Integrity, users and user groups, procedures and triggers, etc.
- Taxa Observations Database
  - Database constraints
  - Database Triggers
- FlyCatcher Radar Database
  - Data Join and Scientific Calculations
  - Users, User Groups, and access rights
- Conclusion

Data Flow and General Information System Components for BAM Applications



Why Databases?

- Complexity of the data in today's applications.
- Requirement of advanced applications:
  - Information needs to be well structured
  - Access to data must be facilitated
  - Private data needs to be properly protected, while public data is made available
  - Access to data must be tailored to the users' authorizations
  - etc.
- Extensibility and expansion of the applications must be supported
  - Extensible data models

Data Modeling: Approach

- Best way to organize the information within organizations is to design their corresponding data models and to chose the appropriate DBMS for their implementation.
  - The data modeling can be carried out by a data architect, while the DBMS is chosen based on the application requirements.
  - Some DBMSs better suit in certain application domains than others
- Data access is facilitated and enforced through:
  - Access mechanisms tailored to the user authorization
  - Ability to access data from different programming environments
  - Providing the necessary access mechanisms, to build client-server applications, using standard tools and middleware solutions (e.g. JDBC, ODBC, and SQLJ)
  - Providing the necessary access mechanisms for specific applications, which require accessing the data from different perspectives (e.g. Java, C, and Perl)

Databases from Design to Implementation

- Database Model
  - A good database model ensure the proper and complete coverage of all the application's metadata.
- Data Sets
  - A valuable data sets are necessary in order to take full advantage of the underlying data model of the database.
- API & Interfaces
  - Proper Interfaces to the Application's database in order to make use of the underlying data and metadata.

Data Integrity

The database system provides a wide range of declarative integrity rules, thus reducing the programming requirements for applications.

- A Key comprises one or more columns and can be defined for each table. The database system ensures that each key is unique.
- By specifying NOT NULL, you ensure that the NULL value is not accepted in individual columns.
- DEFAULT definition defines default values for each column.
- Referential integrity conditions declares deletion and existence dependencies between the rows in two tables.
- Procedures and Triggers implement complex integrity rules that require further access to tables.

Database Users and User Groups

There are four database user classes when the database is operating:

- Database system administrator (SYSDBA)
- Database administrators (DBA users)
- RESOURCE users
- STANDARD users

- RESOURCE users can define data and database procedures and grant other users privileges for these database objects.

- STANDARD users only have access to data and database procedures that were defined by other users and for which they have privileges. STANDARD users themselves can define view, synonyms, and temporary tables.

Database Users and User Groups

Database Users can be grouped into User Groups.

- All database objects defined by members of a certain user group can be identified by the user group name.
- If a member of a user group creates objects, each member of that group can work with these objects as if they owners these object.
- All of the members of a user group have the same rights with regard to data assigned to the group.

User Authorizations, Privileges, and Roles

- User authorizations are the Authorizations of the Database Manager operator (DBM operator) that he/she needs to use the required functionality of the Database Manager.

- A Privilege is used to impose restrictions on operations carried out on certain objects.
  - Users can only execute operations on objects if they have been granted the privileges to do so.

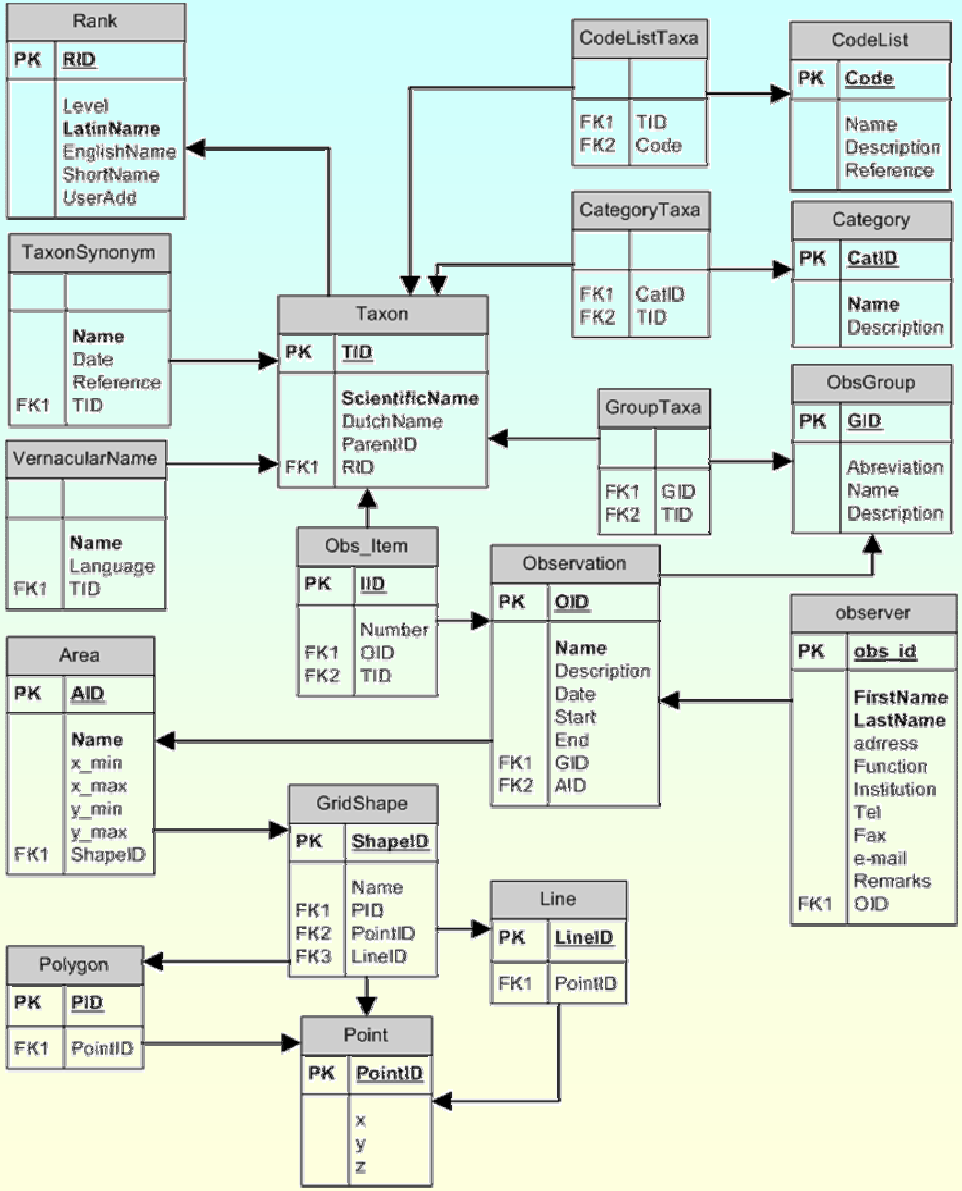
- A Role is a collection of Privileges.
  - Like a privilege, a role can be assigned to a different user.

Database Procedures and Triggers

Database procedures are small programs written in special SQL syntax (PL/SQL) which run on the database server and can be called explicitly from the programming language of the application.

Triggers are special procedures that run implicitly on a base table (or a view table built on this base table) after a data manipulation statement has been executed.

Database Model for Taxa Observations



DDL Schema for Taxa Observations

```
CREATE TABLE Rank (
  RID FLOAT NOT NULL,
  Level Integer,
  LatinName VARCHAR(80),
  EnglishName VARCHAR(80),
  ShortName VARCHAR(80),
  UserAdd VARCHAR(80) DEFAULT USER,
  PRIMARY KEY (RID)
)

CREATE TABLE Taxon(
  TID VARCHAR(10) NOT NULL,
  ScientificName VARCHAR(250),
  DutchName VARCHAR(250),
  Author VARCHAR(250),
  ParentID VARCHAR(10),
  PRIMARY KEY (TID),
  RID FLOAT NOT NULL,
  FOREIGN KEY (RID) REFERENCES Rank
   (RID)
)
```

```
CREATE TABLE VernacularName(
  Name VARCHAR(250) NOT NULL,
  Language VARCHAR(50) NOT NULL,
  PRIMARY KEY (Name, Language),
  TID VARCHAR(10) NOT NULL,
  FOREIGN KEY (TID) REFERENCES Taxon
   (TID)
)

CREATE TABLE TaxonSynonym(
  TSID FLOAT NOT NULL,
  Name VARCHAR(250),
  "Date" TIMESTAMP DEFAULT TIMESTAMP,
  Reference VARCHAR(80) DEFAULT USER,
  PRIMARY KEY (TSID),
  TID VARCHAR(10) NOT NULL,
  FOREIGN KEY (TID) REFERENCES Taxon
   (TID)
)
```

Taxa Observations Database - 1

- Select Taxa with their corresponding French Name:
  - Select T.ScientificName, T.DutchName, T.Author, V.Name FrenchName FROM Taxon T, Vernacularname V where T.TID=V.TID and V.Language='French’
- Select Taxa with CodeList ‘2B’:
  - select T.*, C.Code, C.description from Taxon T, CodeListTaxa CT, CodeList C where T.TID=CT.TID and CT.Code=C.Code and C.Code='2B’
- Select Taxa with Category ‘D’:
  - select T.*, C.CatID, C.description from Taxon T, CategoryTaxa CT, Category C where T.TID=CT.TID and CT.CatID=C.CatID and C.CatID=’D'

Taxa Observations Database - 2

- Create Trigger Rank_Insert

```
CREATE TRIGGER Rank_insert FOR Rank AFTER INSERT EXECUTE
(
  TRY
   update Rank SET Level = Level + 1 where Level >= :NEW.Level
   and RID <> (select max(RID) from Rank where
   Level=:NEW.Level);
  CATCH
  IF $rc <> 100 THEN
   STOP ($rc, 'unexpected error in the Rank Trigger involving
   Level Push');
)
```

  - insert into RANK (RID,LEVEL, LatinName) values (20,15,'Before Species')
  - select * from rank order by level
  - delete from rank where RID = 20

Taxa Observations Database - 3

- Create Trigger Rank_Delete

```
CREATE TRIGGER Rank_delete FOR Rank AFTER DELETE EXECUTE
(
  TRY
   update Rank SET Level = Level - 1 where Level > :OLD.Level;
  CATCH
  IF $rc <> 100 THEN
   STOP ($rc, 'unexpected error in the Rank Trigger involving Push
   Level');
)
```

Taxa Observations Database - 4

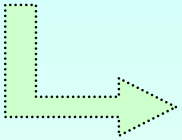
- Create Trigger Taxon_Update

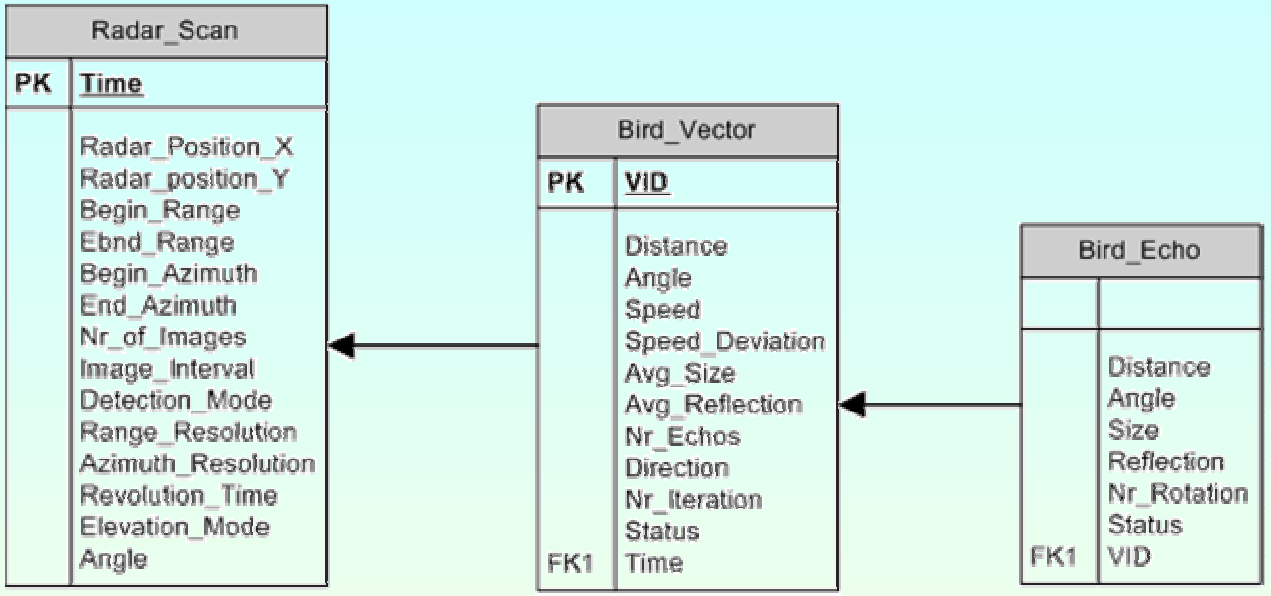
```
CREATE TRIGGER TAXON_UPDATE FOR TAXON AFTER UPDATE EXECUTE
(
  TRY
    IF NEW.ScientificName <> OLD.ScientificName THEN
       INSERT INTO TaxonSynonym(TSID, Name, TID) VALUES ((SELECT
   MAX(TSID)+1 FROM TaxonSynonym), :OLD.ScientificName, (SELECT
   MAX(TID) FROM Taxon WHERE ScientificName=:NEW.ScientificName));
  CATCH
  IF $RC <> 100 THEN
      STOP ($ERRMSG, 'unexpected error in the Taxon Trigger
   involving TaxonSynonym Update');
)
```

- update taxon set ScientificName='New Name for Acherontia atropos' where ScientificName='Acherontia atropos'

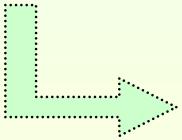
Database Model for Fly Catcher Radar

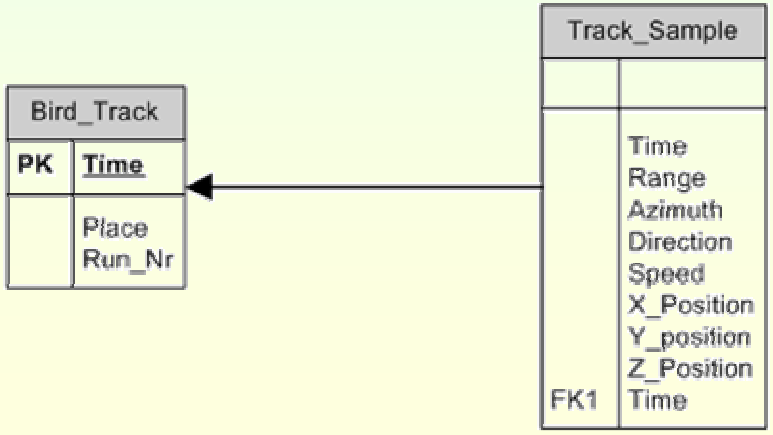
Bird Vectors





Tracked Echos





FlyCatcher Radar Database - 1

- Select Bird_Echos scanned at Radar_Position_X= 187600 and for which the Bird_Vector Speed is greater than 5
  - `select E.* from Radar_Scan R, Bird_Vector V, Bird_Echo E where R."Time"=V."Time" and V.VID = E.VID and V.Speed>5 and R.Radar_Position_X=187600`
- Select Bird_Vectors with their corresponding Height and Horizontal Distance from the Radar Position
  - `select VID, Distance, Angle, distance*sin(angle) Height, distance*cos(angle) H_Distance, Speed, AVG_Size, AVG_Reflection, Nr_Echos, Direction, "Time" Time_Interval from Bird_Vector`
- Select Bird_Vectors with their corresponding X,Y Coordinates calculated based on the Radar Position
  - `select V.VID, (R.Radar_Position_X+V.Distance*COS(V.Angle)) Vector_Pos_X, (R.Radar_Position_Y+V.Distance*SIN(V.Angle)) Vector_Pos_Y, V.Speed, V.Speed_Deviation, V.AVG_Size, V.AVG_Reflection, V.Nr_Echos, V.Direction FROM Radar_Scan R, Bird_Vector V`

FlyCatcher Radar Database - 2

- Create a view corresponding to Bird_Vectors with their corresponding X,Y Coordinates:
  - `Create View VectorsWithCoordinates as select V.VID, (R.Radar_Position_X+V.Distance*COS(V.Angle)) Vector_Pos_X, (R.Radar_Position_Y+V.Distance*SIN(V.Angle)) Vector_Pos_Y, V.Speed, V.Speed_Deviation, V.AVG_Size, V.AVG_Reflection, V.Nr_Echos, V.Direction FROM Radar_Scan R, Bird_Vector V`

- Select VectorsWithCoordinates with speed>5 and Avg_Size<10
  - `select * from VECTORWITHCORDINATES where Speed>5 and Avg_Size<10`

FlyCatcher Radar Database - 3

- Create Users and User Groups with some privileges (an example)
  - `create usergroup group1 resource not exclusive`
  - `create user ammar password ammar usergroup group1`
  - `create user Jelmer password Jelmer not exclusive`
  - `grant select,update on Radar_Scan, Bird_Vector, Bird_Echo, Bird_Track,  Track_Sample to group1`
  - `grant select on Radar_Scan, Bird_Vector, Bird_Echo to Jelmer`

Conclusion

- What to Use: Databases of File System?
- Databases are better since they provide concepts for data structures, integrity constraints, users authorizations, and data access rights.
- Databases also ease the access to data and allows for data model extension
- But, the size of databases may grow very rapidly. The application's manager should then watch the database performance and efficiency.