

QUILLOW BIRD MONITORING INFORMATION SYSTEM

Author: Chandrabali Karmakar

Supervisor : Prof. Dr Alfred Schultz

Overview

2

- This database project deals with breeding bird data coming from a long term monitoring project in the Quillow river catchment area in North East Brandenburg.
- The monitoring was performed in two different periods from 1999 – 2002 and from 2013 – 2015.
- The monitoring project was performed under the leadership of the Leibniz Center for Land Use Research (ZALF) Müncheberg, Institute for Land Use Systems (Dr. M. Glemnitz, Dr. U. Stachow) in cooperation with the University for Sustainable Development (HNEE) (Prof. Dr. A. Schultz)

Overview-The final objective

3

- The overall objective of the underlying bird monitoring project is to derive and test biodiversity indicators based on breeding bird occurrences in order to :
 - describe states and trends in biodiversity development in the considered landscape
 - derive relationships between the occurrence of breeding birds and landscape structure
 - analyze potential reasons for observed changes, e.g. due to land use changes

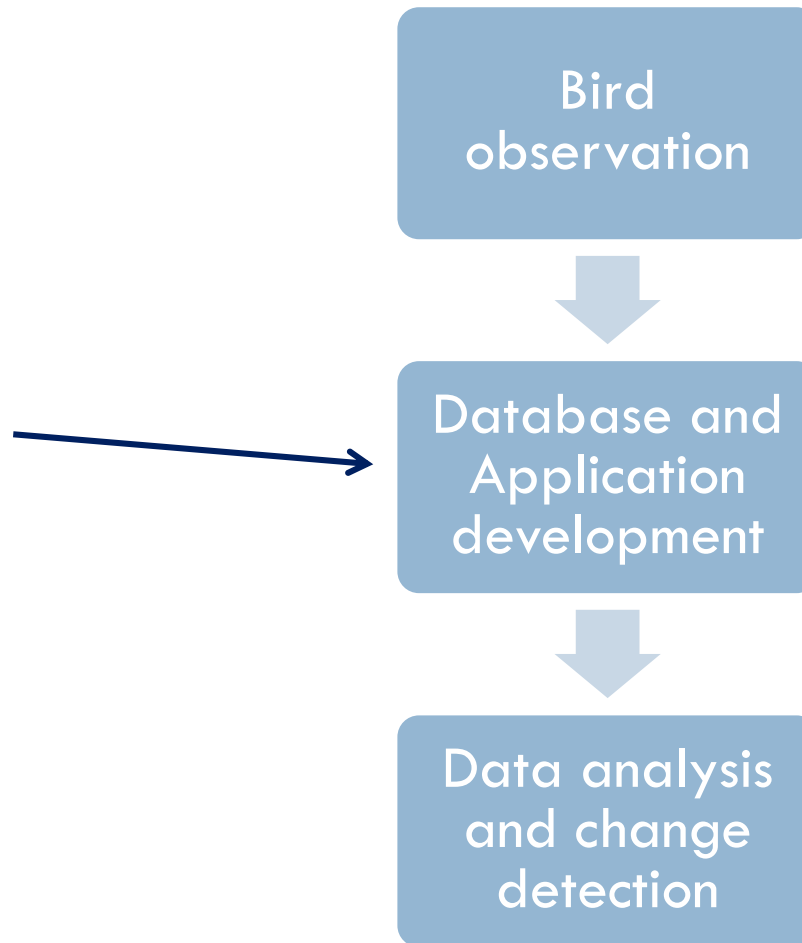
Overview –The current need

4

- The bird occurrence data were listed in an acquisition sheet resulting in more than 20000 records , each record containing 22 values.
- This project aims to develop a database for bird monitoring data and a desktop application for further analysis .

Overview- the big picture

5



Overview- Acquisition sheet

6

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	ID	PUNKT_NIJAHR	BONNR	X	Y	DATUM	UHRZEIT_BE	UHRZEIT_EN	ARTENCODE	VOGELART	CODE_ARTEN	GATTUNG	SPECIES_sh	ANZ_INDI	C_WITTER	WITTERUN	C	
2	1	1.01	1999 99/03/01	5402626	5911378	1999-03-30 12:07:22	12:05	Am	Amsel	550	Turdus	merula	2	2	heiter; ku			
3	2	1.01	1999 99/04/01	5402626	5911378	1999-04-24 12:06:20	10:42	Am	Amsel	550	Turdus	merula	1	7	sonstiges			
4	3	1.01	1999 99/05/01	5402626	5911378	1999-05-27 12:06:05	09:35	Am	Amsel	550	Turdus	merula	1	1	heiter; wa			
5	4	1.01	1999 99/05/01	5402626	5911378	1999-05-27 12:06:05	09:35	Am	Amsel	550	Turdus	merula	1	1	heiter; wa			
6	5	1.01	1999 99/06/01	5402626	5911378	1999-06-17 12:06:10	09:35	Am	Amsel	550	Turdus	merula	1	1	heiter; wa			
7	6	1.01	1999 99/06/01	5402626	5911378	1999-06-17 12:06:10	09:35	Am	Amsel	550	Turdus	merula	1	1	heiter; wa			
8	7	1.01	2000 00/09/01	5402626	5911378	2000-05-03 12:05:30	08:30	Am	Amsel	550	Turdus	merula	3	1	heiter; wa			
9	8	1.01	2000 00/05/02	5402626	5911378	2000-05-18 12:05:30	08:30	Am	Amsel	550	Turdus	merula	1	13	heiter; be			
10	9	1.01	2000 00/05/01	5402626	5911378	2000-06-10 12:05:20	09:20	Am	Amsel	550	Turdus	merula	1	1	heiter; wa			
11	10	1.01	2000 00/05/01	5402626	5911378	2000-06-10 12:05:20	09:20	Am	Amsel	550	Turdus	merula	2	1	heiter; wa			
12	11	1.01	2001 01/09/02	5402626	5911378	2001-05-22 12:06:30	09:55	Am	Amsel	550	Turdus	merula	1	1	heiter; wa			
13	12	1.01	2001 01/06/01	5402626	5911378	2001-06-13 12:07:00	10:15	Am	Amsel	550	Turdus	merula	1	2	heiter; ku			
14	13	1.01	2002 02/03/01	5402626	5911378	2002-03-17 12:06:50	12:00	Am	Amsel	550	Turdus	merula	1	4	bewoelkt;			
15	14	1.01	2002 02/03/01	5402626	5911378	2002-03-17 12:06:50	12:00	Am	Amsel	550	Turdus	merula	2	4	bewoelkt;			
16	15	1.01	2002 02/04/01	5402626	5911378	2002-04-27 12:07:20	10:10	Am	Amsel	550	Turdus	merula	1	4	bewoelkt;			
17	16	1.01	2002 02/05/01	5402626	5911378	2002-05-09 12:06:55	10:00	Am	Amsel	550	Turdus	merula	3	1	heiter; wa			
18	17	1.01	2002 02/05/02	5402626	5911378	2002-05-23 12:07:15	10:15	Am	Amsel	550	Turdus	merula	1	1	heiter; wa			
19	18	1.01	2002 02/06/01	5402626	5911378	2002-06-12 12:07:20	10:20	Am	Amsel	550	Turdus	merula	1	3	bewoelkt;			
20	19	1.02	1999 99/05/01	5402948	5911755	1999-05-27 12:06:05	09:35	Am	Amsel	550	Turdus	merula	1	1	heiter; wa			
21	20	1.02	1999 99/06/01	5402948	5911755	1999-06-17 12:06:10	09:35	Am	Amsel	550	Turdus	merula	1	1	heiter; wa			
22	21	1.02	2000 00/05/02	5402948	5911755	2000-05-18 12:05:30	08:30	Am	Amsel	550	Turdus	merula	1	13	heiter; be			
23	22	1.02	2000 00/06/01	5402948	5911755	2000-06-10 12:05:20	09:20	Am	Amsel	550	Turdus	merula	1	1	heiter; wa			
24	23	1.02	2001 01/06/01	5402948	5911755	2001-06-13 12:07:00	10:15	Am	Amsel	550	Turdus	merula	1	2	heiter; ku			
25	24	1.02	2002 02/03/01	5402948	5911755	2002-03-17 12:06:50	12:00	Am	Amsel	550	Turdus	merula	1	4	bewoelkt;			

Goals

7

- Reliable storage of data
- Efficient data operations for future analysis
- User-friendly desktop application.

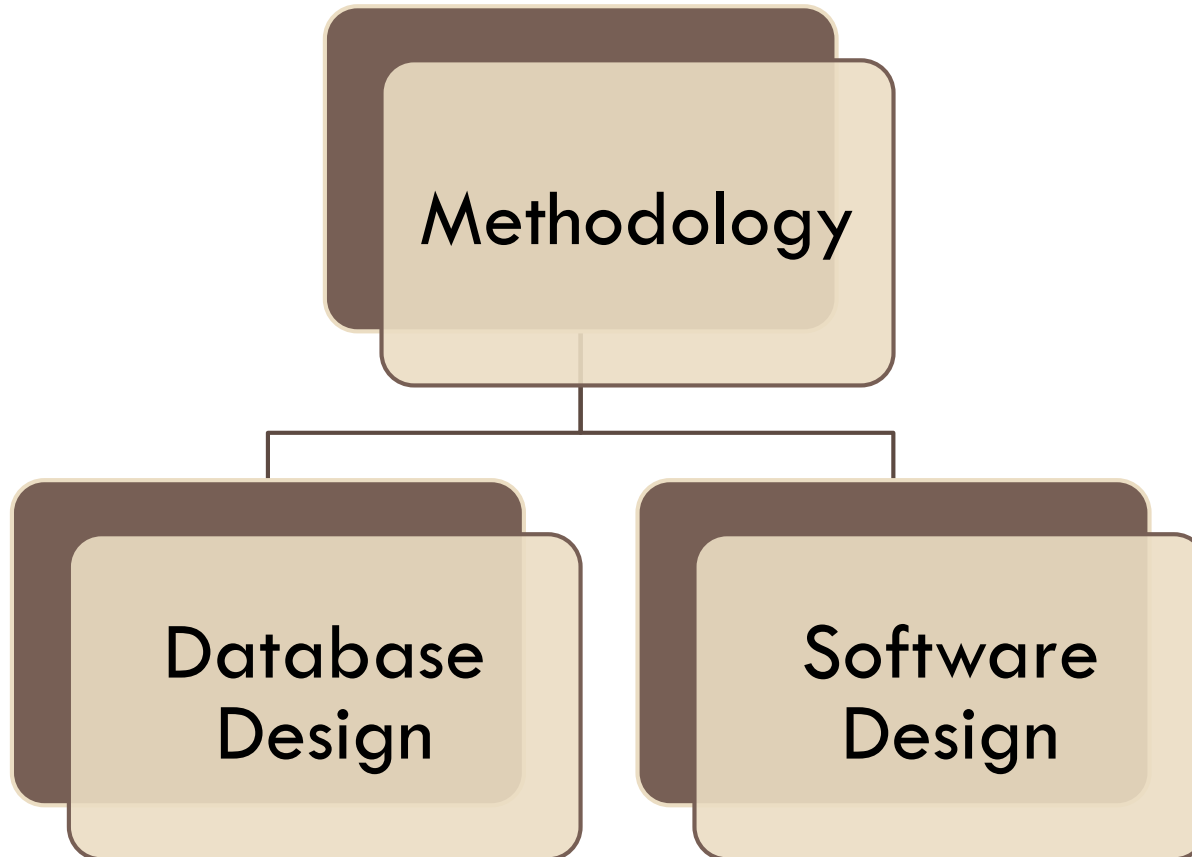
Motivations

8

- Organize the data in a better way to easily answer data questions .
- Some ready-to-see reports may answer some commonly raised queries by users(scientists and others)
- A desktop application can help inserting new observation data, retrieve them and finally help further data analysis.

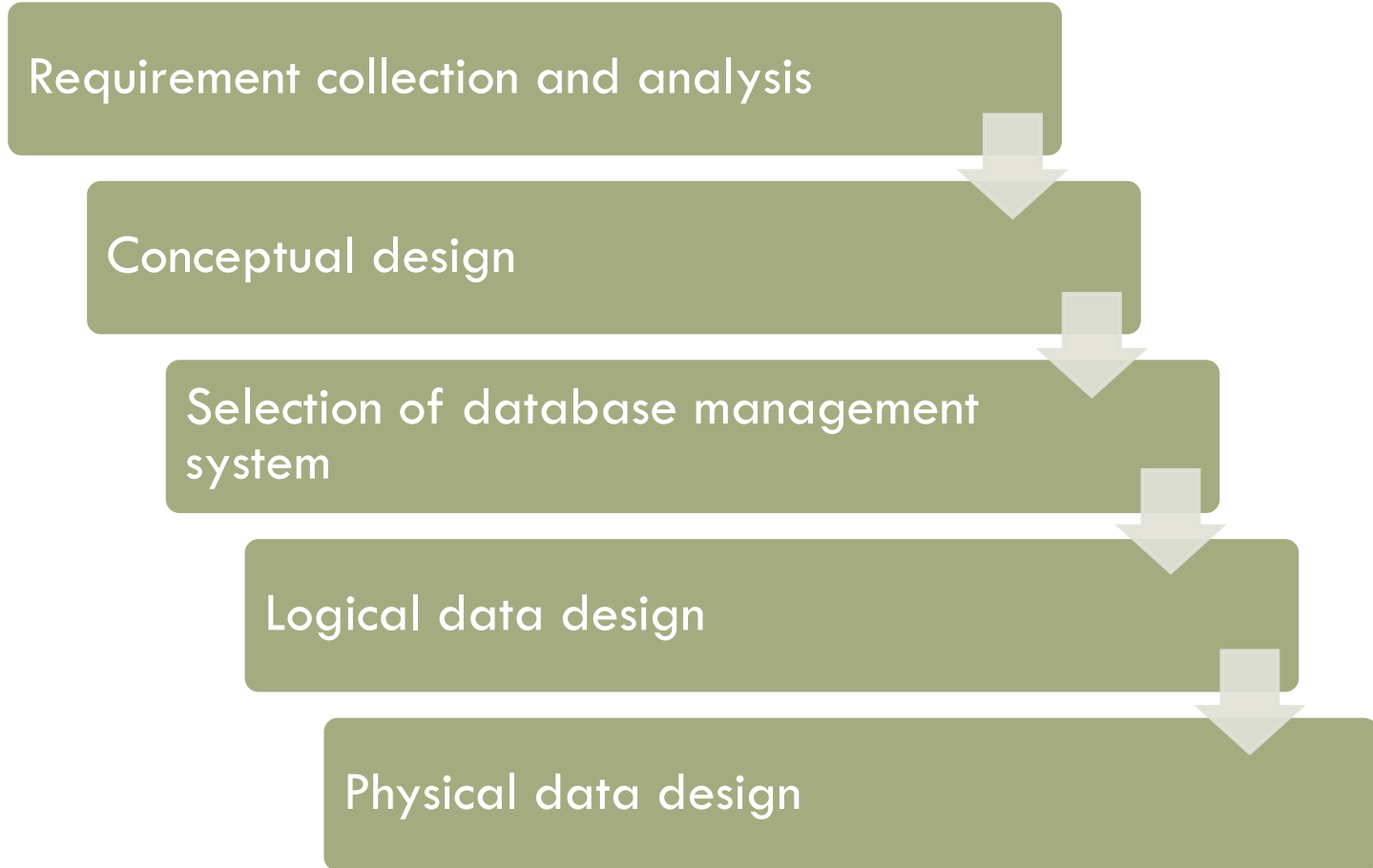
Research Methodology

9



Research Methodology ~ database design

10



Research Methodology ~ database design ~ Requirement collection and Analysis

11

- Who may require data and in which format.
- Which data may be required most often.
- Which data may be not required and thus eligible for elimination.

Research Methodology ~ database design ~ **Conceptual design**

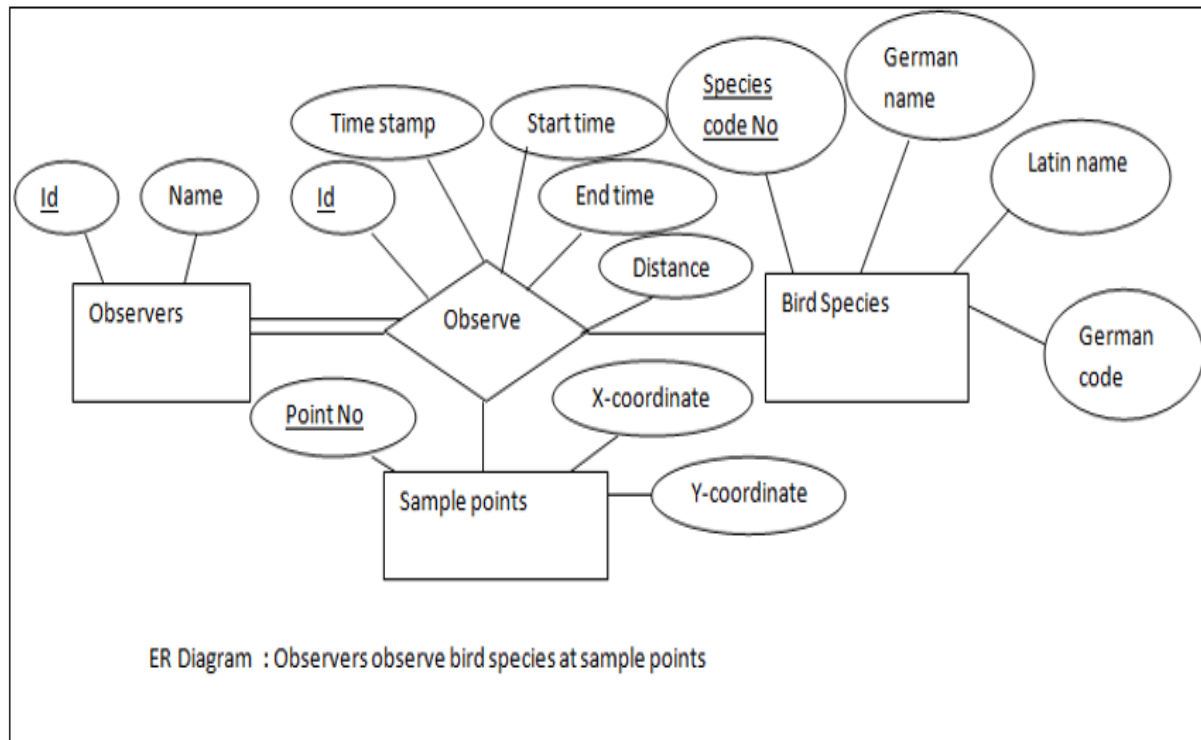
12

- Entity- relationship diagram
- Different entities and their relationships are to identified.
- Entity-relationship diagram is not specific to any database model(relation ,hierarchical etc)
- It is possible to navigate to another database model.

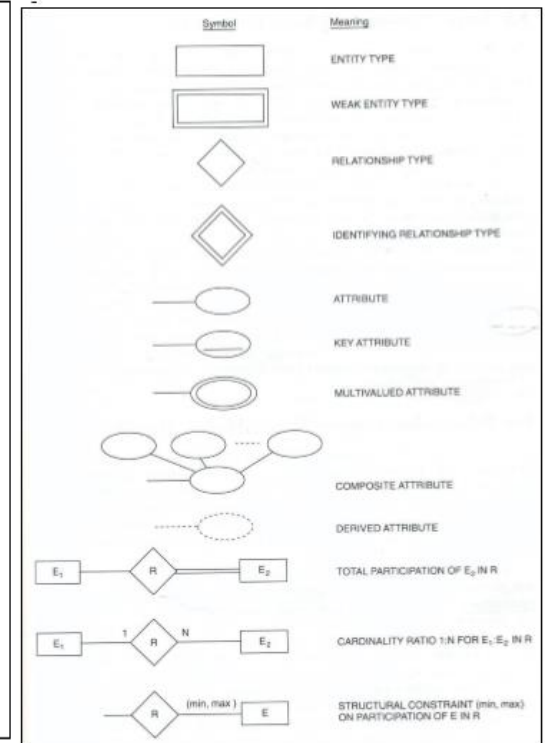
Research Methodology ~ database design ~ Conceptual design

13

Conceptual Design - entity relationship diagram



Conceptual Design Symbols



Research Methodology ~ database design ~ **Selection of DBMS**

14

- Selection of data model.
- Requirements were to consider users with minimal software knowledge.
- Possible to migrate to more powerful DBMS.

Research Methodology ~ database design ~ **Logical data design**

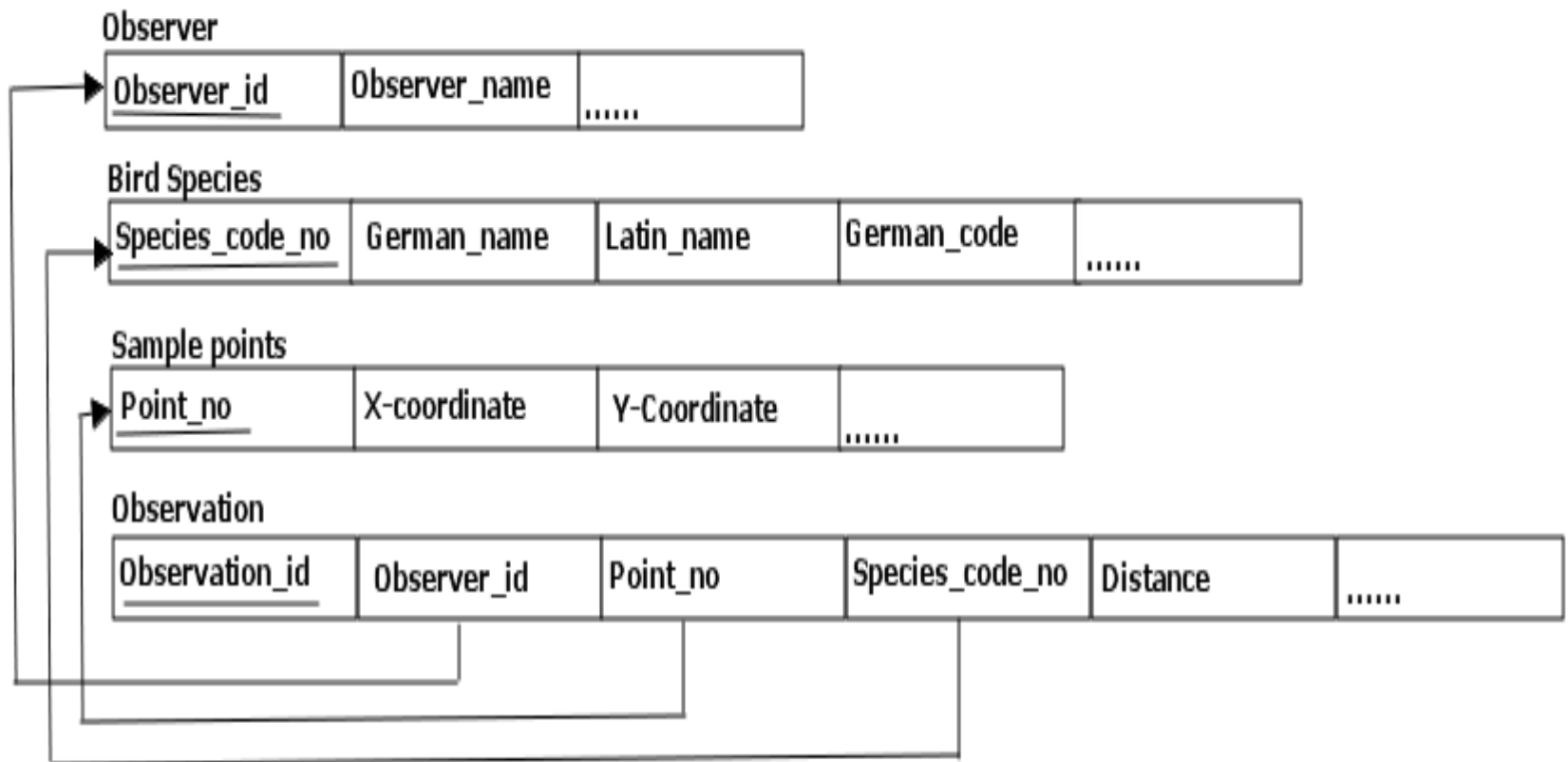
15

- Logical table formation.
- Specific to relational database.
- A pen and paper method, can also be done using special software.

Research Methodology ~ database design

~ Logical data design

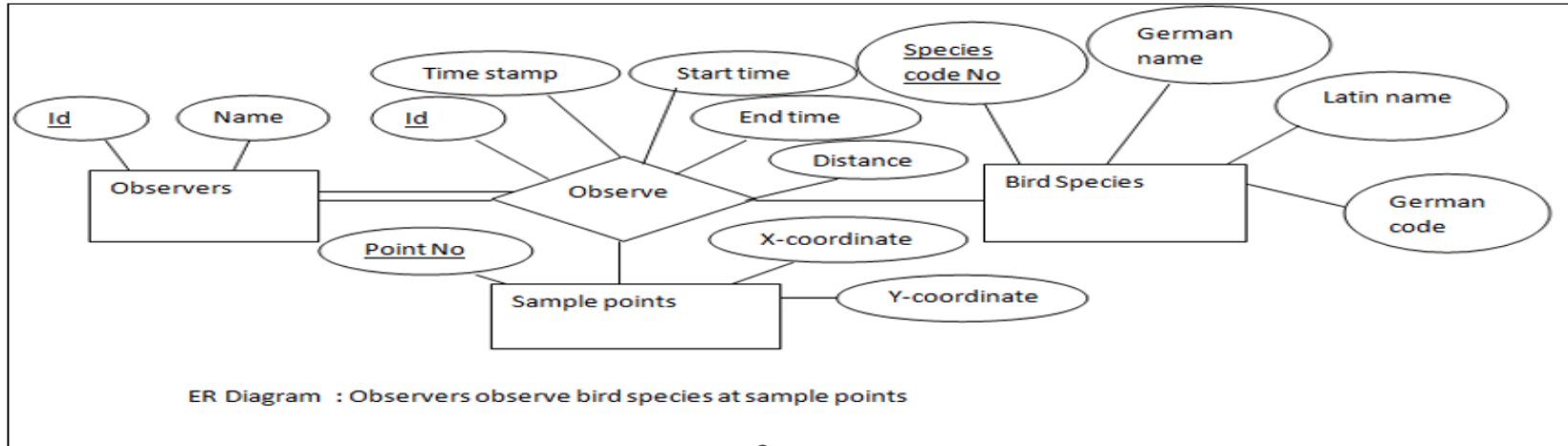
16



Research Methodology ~ database design ~ Logical design (Mapping)

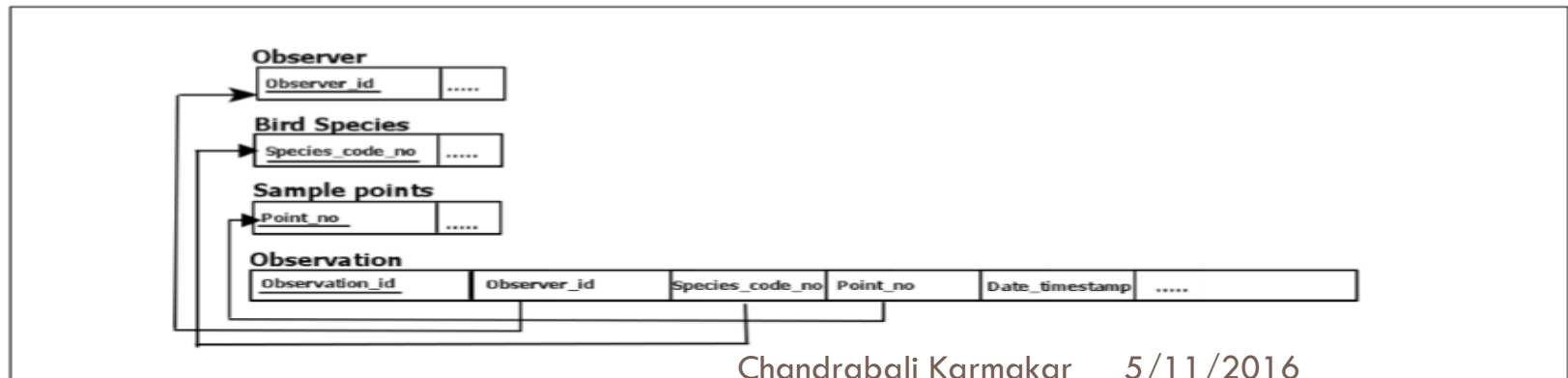
17

Conceptual Design - entity relationship diagram



Mapped to

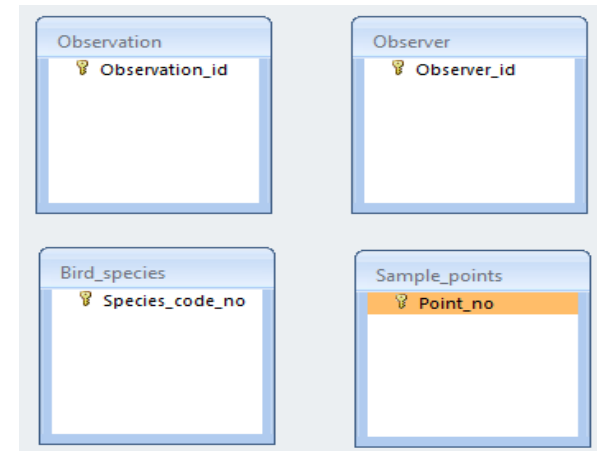
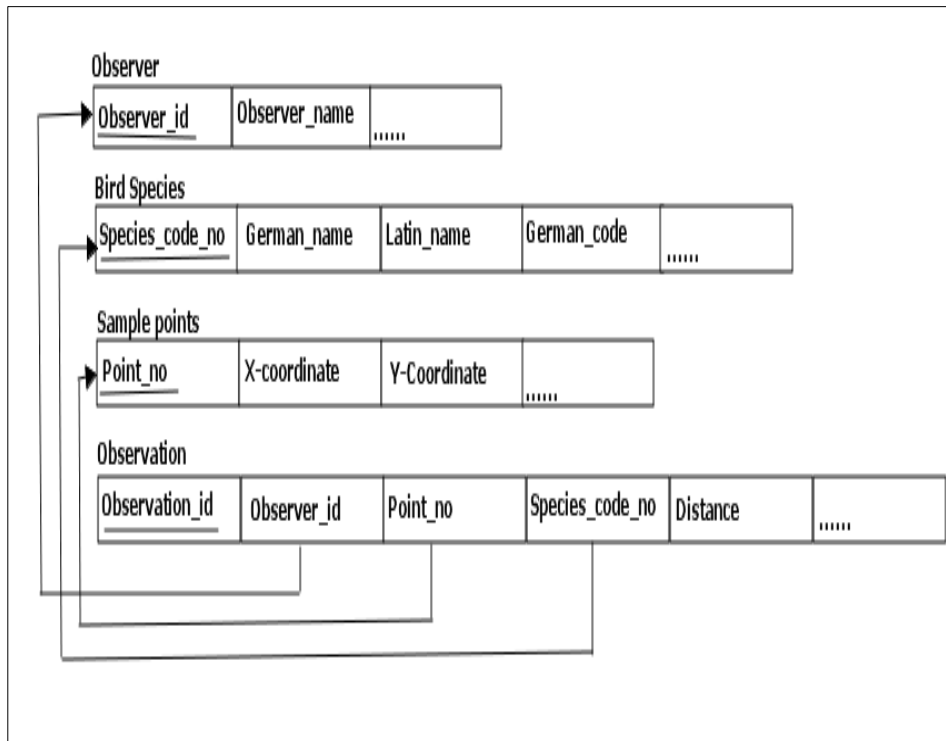
Logical Design - database schema



Research Methodology ~ database design ~ Physical data design 1

18

- Initial table creation by realizing the schema(logical design)



Research Methodology ~ database design ~ Physical data design





19

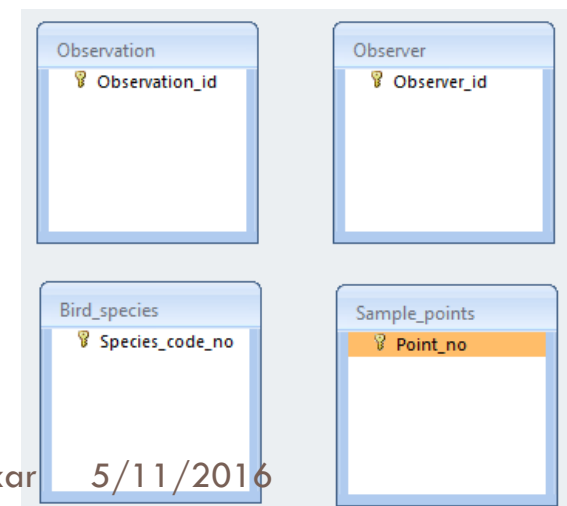
- Step 1: Actual relation formation(table creation)

Columns from acquisition sheet

Column name(German)	Column name(English) and explanation
ID	An identification no of every observation
PUNKT_NR	Point number – sample point number
JAHR	Year of observation
BONR	Date
X	X coordinate of the point
Y	Y coordinate of the point
DATUM	Date timestamp
UHRZEIT_BE	Starting time of the particular observation
UHRZEIT_EN	Starting time of the particular observation
ARTENCODE_	Species code in German
VOGELART	Species name in German
CODE_ARTEN	Species code number
GATTUNG	Genus in Latin
SPECIES_sh	Species name in Latin
ANZ_INDIVI	Number of individuals observed
C_WITTERUN	Weather code
WITTERUNG	weather
C_ENTFERNU	Distance coding
ENTFERNUNG	Distance
BEOBACHTER	Observer
Species	Genus and species in Latin

Decision legend

	Moved to the Observation table
	Moved to the Observer table
	Moved to the Sample_points table
	Moved to the Bird_species table

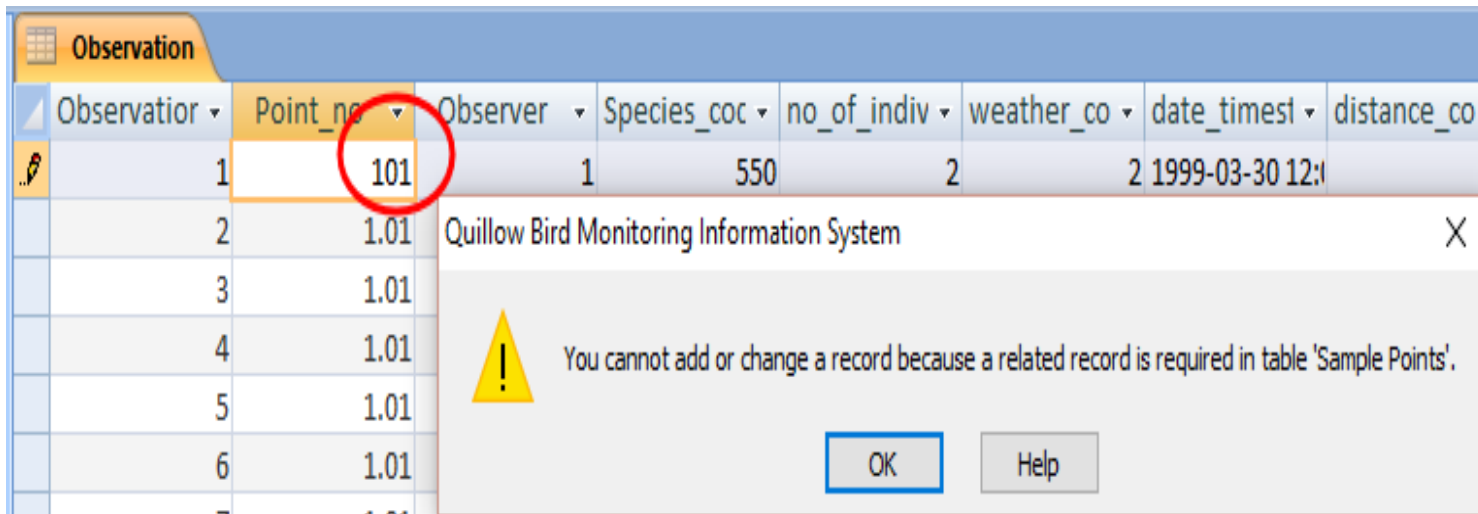


Research Methodology ~ database design

~ Physical data design


20

- Anomalies – leads to inconsistency of data
- Insertion , update, deletion anomaly



Observation	Observator	Point_no	Observer	Species_coc	no_of_indiv	weather_co	date_timest	distance_co
	1	101	1	550	2	2	1999-03-30 12:0	
	2	1.01						
	3	1.01						
	4	1.01						
	5	1.01						
	6	1.01						
	7	1.01						

Quillow Bird Monitoring Information System

 You cannot add or change a record because a related record is required in table 'Sample Points'.

OK Help

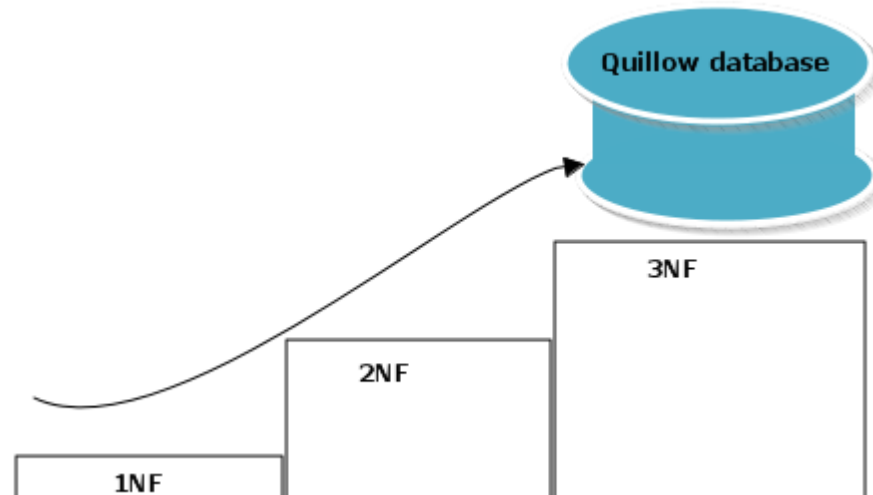
Research Methodology ~ database design ~ **Physical data design**

21

- Normalization -Normalization of data can be considered a process of analyzing the given relation schemas based on their FDs and primary keys to achieve the desirable properties of (1) minimizing redundancy and (2) minimizing the insertion, deletion, and update anomalies.
- This project aims to keep the database in the third normal form.

Research Methodology ~ database design ~ Physical data design

22



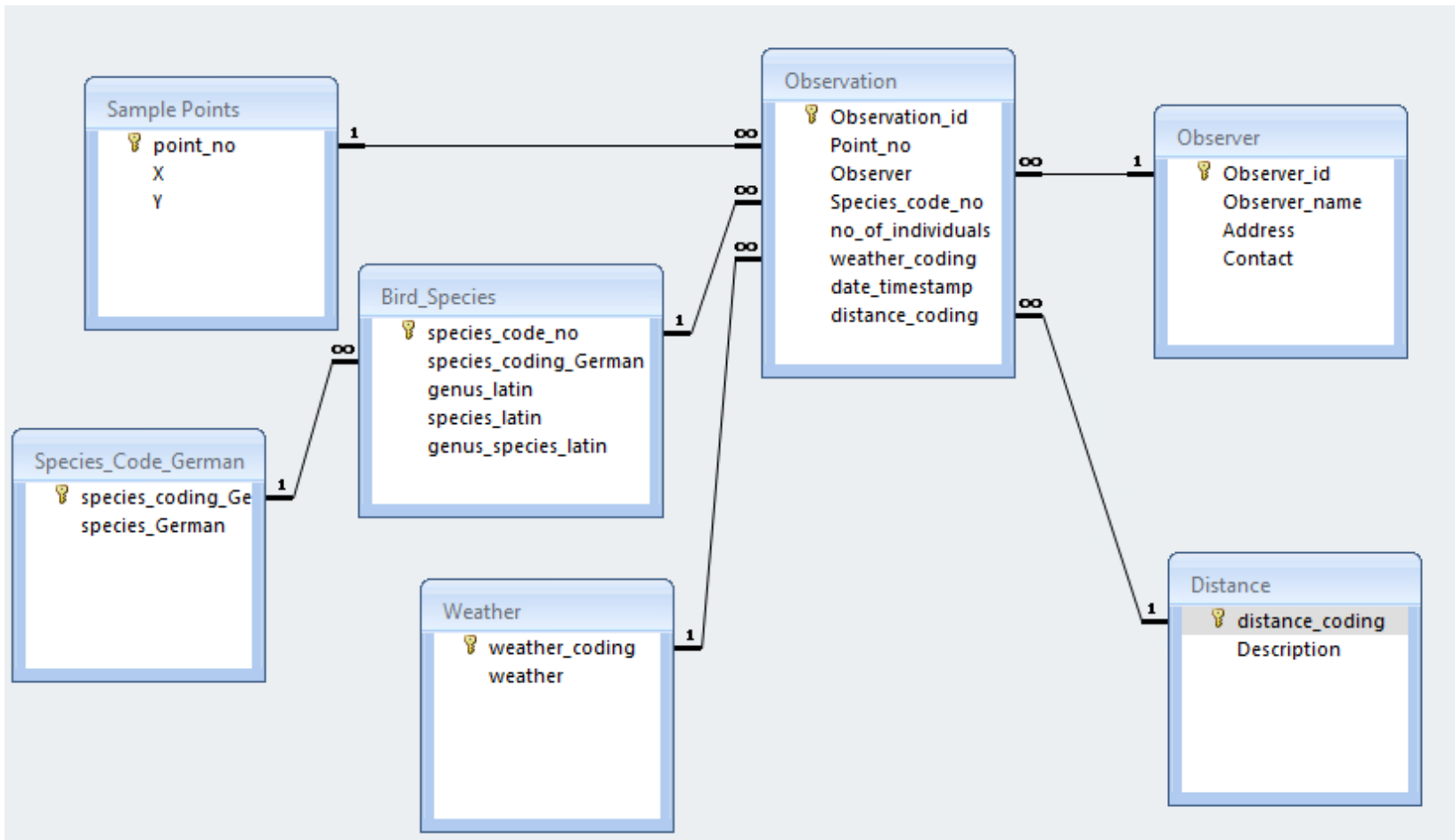
Research Result

23

- A well-designed (satisfies ACID properties) database with minimal redundancy .
- A desktop application that is based on the database.

Research Result - Database

24



Research Result –Application program

25

The screenshot displays a software application window titled "FrmMain". The window is divided into two main sections: "Data Entry Forms" on the left and "Data Query Forms" on the right. Each section has a header bar and a main content area. In the "Data Entry Forms" section, there are two buttons: "Observation Form" (highlighted with a blue border) and "Observer form". In the "Data Query Forms" section, there are two buttons: "Query Observer Form" and "Query Observation Form". Each button is accompanied by a descriptive text block explaining its function.

Data Entry Forms
Please click the proper button for entering data.

Observation Form
This option can be used to store your data about Bird observation.

Observer form
This option can be used to store your data about Bird observation.

Data Query Forms
Please click the proper button for making queries

Query Observer Form
This option can be used to answer your questions about Bird observers.

Query Observation Form
This option can be used to answer your questions about Bird observation.

Research Result –Application program

26

FrmObservation Bird_Species FrmObserver FrmQueryObserverDetails **FrmQueryObservation**

Sample point-weather-year-wise bird occurrence details

Instruction: Enter a year (1999-2002) and sample point no .Please refer to the attached reports and click the submit button to see bird observation details for that year,weather and sample point.

Enter Year:

Enter Sample point no:

Result of your query is shown below:

No. of observations:

No. of species found:

List of species found :

SPECIES CODE NO: 550 LATIN NAME: Turdus merula NO. OF INDIVIDULS OBSERVED: 1
SPECIES CODE NO: 550 LATIN NAME: Turdus merula NO. OF INDIVIDULS OBSERVED: 1
SPECIES CODE NO: 606 LATIN NAME: Fringilla coelebs NO. OF INDIVIDULS OBSERVED: 1
SPECIES CODE NO: 606 LATIN NAME: Fringilla coelebs NO. OF INDIVIDULS OBSERVED: 1
SPECIES CODE NO: 606 LATIN NAME: Fringilla coelebs NO. OF INDIVIDULS OBSERVED: 1
SPECIES CODE NO: 410 LATIN NAME: Alauda arvensis NO. OF INDIVIDULS OBSERVED: 5

Weather-wise bird occurrence

Instruction: Enter a weather code of your choice .Please refer to the attached report Weather Codes and click the submit button to see bird observation details for that weather .

Enter Weather code:

No. of Species found:

Done

Concluding remarks and future scope

27

- Both the database and application program can be extended for better features.
- The application program is being modified to generate input for statistical analysis.

References

28

- Database concepts – Navathe 2000
- A Relational Model of Data for large shared database – E F Codd
- “Landscape-related analysis of breeding bird monitoring data in the case study area Quillow”- Friederike Borges

Thank you for your attention

Questions are welcome



Appendix-1

Results:

The result of normalization is shown below:

Table 3: Relations after normalization process

Old Relation [attributes names are inside paranthesis ; Underlined attributes are key (Prime) attributes]	Functional Dependencies	New Relations after decomposition	Functional Dependencies		Comment						
Bird Species (<u>species_code_no</u> , <u>species_coding_German</u> , species_German, genus_latin, species_latin, genus_species_latin)	species_code_no → species_coding_German species_code_no → genus_latin species_code_no → species_latin species_code_no → genus_species_latin species_coding_German → species_German	Bird_Species (species_code_no, species_coding_German, genus_latin, species_latin, genus_species_latin) Species_Code_German (species_coding_German, species_German)	<table border="1"> <thead> <tr> <th data-bbox="1103 736 1238 839">Relation</th> <th data-bbox="1238 736 1489 839">Functional Dependencies</th> </tr> </thead> <tbody> <tr> <td data-bbox="1103 839 1238 1082">Bird_Species</td> <td data-bbox="1238 839 1489 1082"> species_code_no → species_coding_German species_code_no → genus_latin species_code_no → species_latin species_code_no → genus_species_latin </td> </tr> <tr> <td data-bbox="1103 1082 1238 1130">Species_Code_German</td> <td data-bbox="1238 1082 1489 1130"> species_coding_German → species_German </td> </tr> </tbody> </table>	Relation	Functional Dependencies	Bird_Species	species_code_no → species_coding_German species_code_no → genus_latin species_code_no → species_latin species_code_no → genus_species_latin	Species_Code_German	species_coding_German → species_German		All relations are in 3NF now
Relation	Functional Dependencies										
Bird_Species	species_code_no → species_coding_German species_code_no → genus_latin species_code_no → species_latin species_code_no → genus_species_latin										
Species_Code_German	species_coding_German → species_German										

Appendix-2

31

Screenshot 6 –Null value handling

```
TxtAddress.SetFocus
If IsNull(rst.Fields(2).Value) Then
TxtAddress.SetFocus
TxtAddress.Text = "Not Available"
ElseIf (Not IsNull(rst.Fields(2).Value)) Then
TxtAddress.SetFocus
TxtAddress.Text = rst.Fields(2)
End If
```

Screenshot 6- Exception handling

```
'the error handler
Error_MayCauseAnError:

If Err.Number = 3021 Then
MsgBox ("Sorry! Please enter a valid id for which we have an existing record !")
ElseIf Err.Number <> 0 Then
MsgBox ("Some error occured." & "Error number: " & Err.Number & ". Error description: " & Err.Description)
End If
```

Screenshot 7 – Data insertion into table

```
Private Sub BtnAdd_Click()
'add data to the table
CurrentDb.Execute " INSERT INTO Observer (Observer_id,Observer_name,Address,Contact) " & _
" VALUES(" & CInt(Me.TxtObsvr_Id) & ",'" & Me.TxtObsvr_Name & "','" & Me.TxtObsvr_Address & "','" & Me.TxtObsvr_Contact & "'" )
```