**Advanced Database**

**Name: Ojaswi Shrestha**

**University ID: C7466889**

**L6 7th Semester**

**Table of Contents**

Data Warehouse

Importance of data warehouse

Features of data warehouse

Approaches of data warehousing

Merits and Demerits of data warehouse

Assignment 1 Part 1

Stakeholder, Use case, Associated KPI, Reports to Evaluate KPI

**Task 1** Data Mart star schema design

**Task 2** Star Schema set up

**Data Warehouse**

Data Warehouse is also known as enterprise data warehouse where data from multiple sources is integrated into a central repository which helps with business intelligence (BI) activities and analytics providing meaningful insights for the development of a business. It is different from a normal DBMS as DBMS stores real time everyday data whereas in data warehouse there is always past and historical data from which analysis could be done.

**Importance of Data Warehouse**

In every business, data warehouses are required to store large vital data from multiple data sources which provide a main source of historical data assisting in faster analytics and decision making. Through one large main data warehouse, data from multiple sources can be overviewed easily and in faster way with fewer errors as it uses ETL process to extract, transform and load data from multiple sources into one data warehouse.

**Features of Data Warehouse**

Data warehouses possess multiple features such as:

* **Subject-oriented**: As it focuses on topics like sales, marketing etc. to provide analysis of them.
* **Integrated**: As it consists of data from multiple sources in a warehouse providing accurate and faster decisions.
* **Non-volatile**: As the data in data warehouse never gets deleted only inserted overtime which stores historical data never losing it.
* **Time-variant**: As the data gets stored in varying time periods either weekly, monthly or yearly, securing the state of data at certain point of time providing long term analysis.

**Merits and Demerits of Data Warehouse**

Data warehouse racks up multiple merits as follows:

* Provides centralized data storage system.
* Better decision making with combined historical data.
* Amplifies business intelligence and analytics.

With its merits, data warehouse also has its demerits as follows:

* More cost and resources due to application and maintenance of data warehouse.
* Causes problems due to unstructured data from other sources.
* ETL process for data warehouses from multiple sources is extremely complex and time consuming.

**Various Approach of Data Warehousing**

**Kimball Approach**

Designed by Ralph Kimball, Kimball approach of data warehousing consists of set of data marts (dimensional tables) then combined into a central data warehouse (fact table). It is a bottom-up development approach which uses star and snowflake schema. It has faster implementation techniques and is convenient for end users to utilize but can cause trouble later on when system grows, which leads to complex integration. This approach is best for small to medium size firms which requires quick BI response.

**Inmon Approach**

Designed by Bill Inmon also known as **“The father of data warehousing”**, Inmon approach of data warehousing central data warehouse is created first and then data marts (dimensional tables) are created from that warehouse. It is a top-down development approach and requires third normalization form for data warehouse design. This approach is best for large firms and has better data quality and consistency but also takes longer to develop the warehouse and to produce results for user.

**Difference between Kimball and Inmon Approaches**

|  |  |
| --- | --- |
| **Kimball** | **Inmon** |
| It is a bottom-up approach: creating data marts first | It is a top-down approach: creating data warehouse first |
| Denormalized form so, simple and easy to understand | Normalized form so, less data redundancy but hard to understand |
| Best for smaller firms, orgs requiring fast analysis and BI | Best for organizations requiring stable and wide reporting system |
| It is faster to deploy as data marts are developed quickly | It is slower to deploy as central warehouse takes longer to built |

**Types of Schemas**

**Star schema**

Star schema is a data model used in data warehousing which includes a central fact table connected to multiple dimension table similar to a star shape. The fact table consists of foreign keys from multiple dimension tables and measures which represent the business metric to be analyzed. Dimension tables consist of surrogate key, natural key and its attributes, connecting to fact table with one-to-many relationship, forming a star like shape. This model is easier to understand and has better query performance with less joins but may cause little data redundancy.

**Snowflake Schema**

Snowflake schema is a data model used in data warehousing which includes a central fact table connected to multiple dimension tables and those dimension tables can be further connected to additional related dimension tables. Those secondary dimension tables do not correct directly to fact table creating a ‘**snowflake**’ like model. This model reduces the redundancy by splitting dimensions into smaller tables but also increases query complexity due to multiple joins.

**Star flake Schema (Hybrid Schema)**

Star flake schema is a mixed design of star and snowflake schema where some dimension tables are denormalized (star schema) or partially normalized (snowflake schema). The structure of the schema is a blend of star and snowflake where fact tables connected to dimension tables or the dimension tables are expanded into further sub-dimensions. This model helps with faster queries and better organization of complex dimensions with less redundancy.

**Difference between Star and Snowflake Schema**

|  |  |
| --- | --- |
| **Star Schema** | **Snowflake Schema** |
| It has simpler structure | It has more complex structure |
| It has faster query performance | It has slower query due to multiple joins |
| It is easier to maintain | It is difficult to maintain |
| It can have high amount of redundancy | It has lower amount of redundancy |

**Why prefer star schema rather than snowflake for Police Crime Data?**

Police crime data has analytical data rather than transaction which leads to simpler queries, low number of joins and faster reporting. ETL or Staging becomes easier and cleaner as well as BI tools like Tableau are best for star schemas. Dimensions are simple with no requirement of normalization and with its simple model, analysis and reporting will be faster and efficient.

**Difference between OLTP and OLAP**

|  |  |
| --- | --- |
| **OLTP** | **OLAP** |
| OLTP stands for Online Transaction Processing | OLAP stands for Online Analytical Processing |
| It handles real time transaction processing with often insert and update using normalized databases. | It supports analytical queries on large historical datasets using denormalized model for reports and aggregation |
| It supports INSERT, UPDATE, DELETE statements | It supports complex SELECT queries |
| It stores current detailed data | It stores high volume of historical data |
| Example: bank transaction, online shopping etcetera | Example: forecasting reports, sale trend analysis etcetera |

**Assignment Part 1**

Specify your stakeholder, use case, associated KPI(s) and the reports to evaluate the KPI.

**Stakeholder**

The primary stakeholders for this police record data mart are “**Police Department**” and “**Crime Analysts**”. The users require accurate, timely and well-structured data to understand and recognize the crime patterns, support their operational decision making and allocate their resources accurately. The data mart helps them to analyze trends on the basis of crime types, their location, time with the case status in a quicker way.

**Use Case**

The use case of the data mart focuses on allowing the police department to observe the frequency of various kinds of crime types across different time periods and locations. By analyzing crime frequency data, the department can recognize crime hotspots, seasonal variations and emerging patterns. The data mart improves planning and efficient deployment of officers to areas where a certain type of crime occurs the most.

**Associated Key Performance Indicators (KPI)**

The chief KPI used in this project is **Crime Type Frequency**. This KPI calculates the number of crimes recorded for each crime type fragmented by time as day/month/year, location and status as open/close. Tracing this KPI allows the department or analysts to understand which crime type occurs the most, how often it occurs and how they differ across different locations and time periods.

**Reports to evaluate the KPI**

Multiple analytical reports can be generated from using the Crime Type Frequency KPI. Some of them are as follows:

* “**Amount of Crime Occurrences by Type per month/year**”: This report sums up crime counts for each crime time on monthly/yearly basis. It helps to identify which crime types are increasing or decreasing over time.
* “**Open VS Closed crimes per location**”: This report compares the number of open and closed cases across different locations. It helps to evaluate productivity and recognize locations where investigations may be delayed.
* “**Trend of Crime Types across Locations**”: This report examines the crime trends across multiple locations. It highlights long-term changes in crime and helps police department to plan strategic responses.
* **“Crime Pattern by Day of Week”**: This report shows which day of the week might have highest number of crimes. It will help arrange the shift schedules for staff.
* **“Crime Resolution Rate by Type”**: This report shows the crime solve rate according to crime type like burglary, fraud etc. It identifies the difficulty of solving a certain crime type.

(Closed Cases / Total Cases) \* 100

**Task 1: Data Mart (DM) star schema design for your chosen use case and KPI(s)**

Create a BASIC star schema to support your use case.

* Document the star schema (SS) design model to support at least one of these reports – QSEE could be used for this 3.5

Star schema of Crime Type Frequency that supports the\* use case of “**Number of Crimes by Crime Type per Month**”

A computer screen shot of a diagram

AI-generated content may be incorrect.

Fig 1. Star schema data mart of Crime Type Frequency

**Overview of the Star Schema**

The star schema made for the use case of Crime Type Frequency supports probe of crime frequency along with multiple dimensions like crime type, location, time and status. The schema has a central fact table surrounded by four-dimensional tables which allow fast aggregation, filtering and trend analysis.

The selected report **“Number of Crimes by Crime Type per Month”** has to fragment crime counts by:

* Crime Type
* Time (Month / Year)
* Location (optional)
* Crime Status (optional)

The star schema provides a simplified, analysis-friendly structure to support this report and other related KPIs.

**Star Schema Components**

**1. Fact Table: FACT\_CRIME\_TYPE\_FREQUENCY**

This fact table consists of the measurable event: **a crime occurrence/count**.  
Each record means one crime summarized by its type, time, location, and resolution status.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| fact\_crime\_type\_id | Surrogate primary key for fact table |
| crime\_count | Measure (represents one crime) |
| reported\_crime\_id\_original | Id of the crimes from both data sources |
| is\_closed\_flag | Represent whether case is closed (1=T, 0=F) |
| is\_escalated\_flag | Represent whether case is in escalated state or not(1=T, 0=F) |
| fk\_time\_id | Links to DIM\_TIME table |
| fk\_location\_id | Links to DIM\_LOCATION table |
| fk\_status\_id | Links to DIM\_STATUS table |
| fk\_crime\_type\_id | Links to DIM\_CRIME\_TYPE table |

**Objective:**

* Allows aggregating crime counts by month.
* Allows filtering by crime type, location, or status.
* Supports crime type trend reports across time.

**2. Dimension Tables**

**DIM\_CRIME\_TYPE**

Defines categories of crimes.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| crime\_type\_id | Surrogate key |
| crime\_type\_id\_original | Id of the crime type from both data sources |
| crime\_type\_description | Description of the crime time |

**DIM\_TIME**

Represents the date for each crime.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| time\_id | Surrogate key |
| full\_date | Full date of the registered crime |
| day | Day of the month |
| month | Month name like Jan, Feb… |
| year | Year (YYYY) |

This dimension enables grouping crimes by month, quarter, and year.

**DIM\_LOCATION**

Derived from REGION, LOCATION and PL\_AREA tables.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| location\_id | Surrogate key |
| area\_id\_original | Area id from pl\_area |
| area | Area name |
| region\_id\_original | Region id from region |
| region | Region name |
| station\_id\_original | Station id from pl\_station |
| station | Station name |

This structure helps analysts to find crime patterns on geographical basis.

**DIM\_STATUS**

Defines the state of each crime either open or closed.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| status\_id | Surrogate key |
| status\_description | Case condition either open or closed |

This supports filtering reports by case status.

**How the Star Schema Supports the Report**

Report Selected: **“Number of Crimes by Crime Type per Month”**

Required fields:

* Crime Type
* Month
* Count of Crimes

**How the star schema supports it:**

* DIM\_TIME provides **month, year** → enables monthly aggregation.
* DIM\_CRIME\_TYPE groups crimes into analysis-friendly categories.
* FACT\_CRIME\_TYPE\_FREQENCY stores a measure field (**crime\_count**) used for counting incidents.
* The schema allows filters on:
  + Location (optional for geographic breakdown)
  + Status (optional)

**Summary**

The star schema is simple but fully supports crime analysis.  
It aligns perfectly with the KPI:

**Crime Type Frequency KPI:**

“**Number of crimes per crime type per time period**”

The schema allows:

* Monthly analysis
* Crime type comparison
* Trend reporting
* Optional filtering by location and case status
* Use the data dictionary template from tutorials (or other) to document a star schema data model for the project, include data quality and transformations.

A table with text on it

AI-generated content may be incorrect.

A table with text on it

AI-generated content may be incorrect.

A close-up of a white box

AI-generated content may be incorrect.

A close-up of a white box

AI-generated content may be incorrect. A close-up of a chart

AI-generated content may be incorrect.

* Select one of the reports\* you have suggested. Illustrate the expected data in the star schema to support the report - use Excel (or oracle or similar) to do this and add a few rows.

Some of the reports to support the star schema includes “**Number of crimes Escalated per Year**”

Escalation Rate by year the “Escalation Rate by Station (%)” line chart visualizes how effectively each police station handles reported crimes over time. By plotting yearly escalation percentages for each station, this chart highlights performance improvements or declines, making it easy to spot stations with rising escalation problems or those consistently performing well. This helps decision-makers identify where additional support, supervision, or training may be needed. The year-to-year trend allows comparison of operational efficiency across different stations and supports strategic planning by revealing workload pressure points. This chart directly supports the report’s aim of analyzing crime handling performance by showing escalation trends, a key KPI linked to investigation quality and case management.

Query:

SELECT

s.station\_name,

TO\_CHAR(rc.date\_reported, 'YYYY') AS year,

COUNT(CASE WHEN rc.crime\_status = 'ESCALATE' THEN 1 END) AS escalated,

COUNT(\*) AS total\_cases,

ROUND( COUNT(CASE WHEN rc.crime\_status = 'ESCALATE' THEN 1 END) / COUNT(\*) \* 100, 2 ) AS escalation\_rate

FROM pl\_reported\_crime rc

JOIN pl\_station s ON rc.fk2\_station\_id = s.station\_id

GROUP BY s.station\_name, TO\_CHAR(rc.date\_reported, 'YYYY')

ORDER BY year;

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**Task 2: Star Schema set up (Data Mart) [2 marks]**

Create a script to implement the tables for the BASIC star schema.

You could use QSEE or AI for this task. *Include the script as an appendix in your upload (name it SS\_DDL.sql)*

***Script reference to SS\_DDL.sql***

A screenshot of a computer

AI-generated content may be incorrect.

Fig 2. Detail view of star schema sql

A screenshot of a computer

AI-generated content may be incorrect.

Fig 3. Detail view of star schema sql

A screenshot of a computer

AI-generated content may be incorrect.

Fig 4. Detail view of star schema sql

A screenshot of a computer

AI-generated content may be incorrect.

Fig 5. Detail view of star schema sql

**Task 3: Data Integration and Extract, Transform and Load (ETL) script to populate the Star Schema (DM) with data [20 marks]**

**Task 3.1 – Data Integration**

Integrate the data from the PS\_wales into the PRCS system (set up by PRCS.sql). Document your decisions and *upload the script as an appendix in your upload (name it data\_integration.sql)*

***Script reference to data\_integration\_staging\_tables***

**Creation of Staging tables**

A screenshot of a computer

AI-generated content may be incorrect.

Fig 6: Detail view for staging tables (1)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 7: Detail view for staging tables (2)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 8: Detail view for staging tables (3)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 9: Detail view for staging tables (4)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 10: Detail view for staging tables (5)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 11: Detail view for staging tables (6)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 12: Detail view for staging tables (7)

**Data Integration in Staging tables**

***Script reference to data\_integration\_data\_load\_in\_staging.sql***

A screenshot of a computer

AI-generated content may be incorrect.

Fig 13: Detail view of data integration in staging tables (1)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 14: Detail view of data integration in staging tables (2)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 15: Detail view of data integration in staging tables (3)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 16: Detail view of data integration in staging tables (4)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 17: Detail view of data integration in staging tables (5)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 18: Detail view of data integration in staging tables (6)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 19: Detail view of data integration in staging tables (7)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 20: Detail view of data integration in staging tables (8)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 21: Detail view of data integration in staging tables (9)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 22: Detail view of data integration in staging tables (10)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 23: Detail view of data integration in staging tables (11)

A screenshot of a computer

AI-generated content may be incorrect.

Fig 24: Detail view of data integration in staging tables (12)

**Creation of Good and bad tables with process and error log**

Script reference from ***Good\_Bad\_tables.sql***

A screenshot of a computer

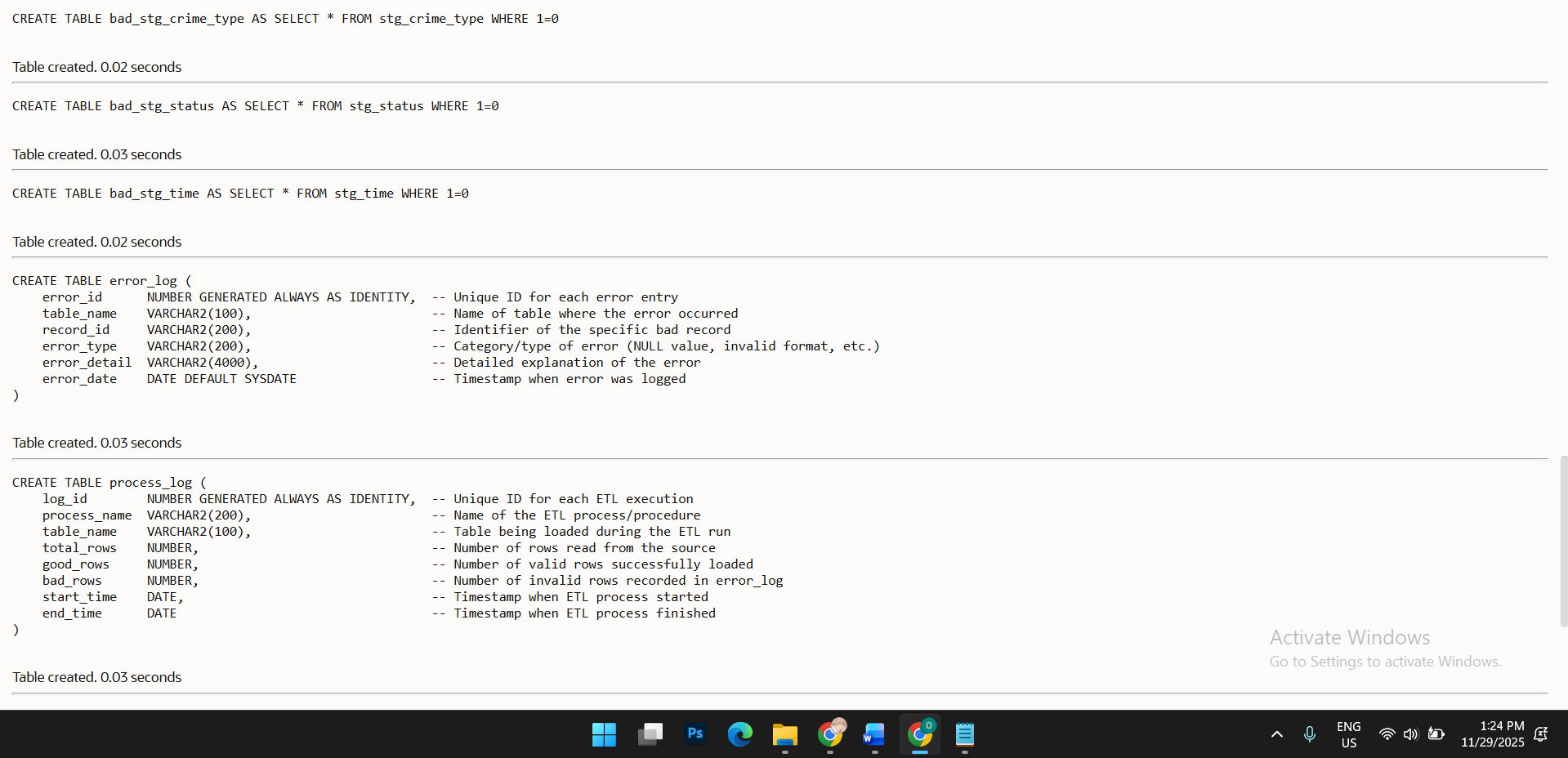
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

**Separating good and bad data into their respective tables with their process and error log**

Script reference from ***Separating good and bad data into tables.sql***

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Cleaning the bad data from the tables and loading them in good tables and deleting the records in bad tables**

Script reference from ***Clean bad data and loading in good table.sql***

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**APPENDIX**