An Online Auction House Database Model

A Project
Submitted to the Department of Computer Science, College of science, University of Baghdad in a Partial Fulfillment of the Requirements for the Degree of B. Sc. in Computer Science

By
Sara Samir Rasheed
Lina Mohammed A.Sultan

Supervised By
Dr. Mehdi G. Duaimi

2011 AD
بِسْمِ ﷲ الرَّحْمَنِ الرَّحِيمِ
اقْرَأْ بِاْسْمِ رَبِّكَ الَّذِي خَلَقَ خَلَقَ الإِنسَانَ مِنْ عَلَقٍ
إِنَّا نَقْرَأُ وَرَبُّكَ الَّذِي عَلَمَ بِالْقَلَمِ عَلَمَ
الْإِنسَانَ مَا لَمْ يَعْلَمَ
صدق ﷲ العلي العظيم
سورة العلق (1-5)
الإهداء

الي من جعلا نفسيهما شمعة تحترق لتنير دربي ... امي وابي.

الي من كانوا يضيفون لي الطريق ويساندوني ويتنازلون عن حقوقهم
لأرضاني والعيش في هناء ... اخواني واخواتي وصديقاتي.

الي من وقفوا على المنابر واعدوا من حصيلة افكارهم لينيروا دربنا
وقبل ان نمضي نقدم اسمى آيات الشكر والامتنان والتقدير والمحبة...

الي الذين حملوا اقدس رسالة في الحياة ...
الي الذين مهدوا لنا طريق العلم و المعرفة ...
الي جميع استذاتتي الافاضل ...

وخاصة الأستاذ المشرف د.مهدي گزار.
Abstract

In this project, an online auction house system is presented. It has a very large potential market of sellers and buyers. An On Line Transaction Processing (OLTP) database model structure is, therefore, desirable. The project involves the design and implementation of an online auction house system. The analysis stage is performed first for the case study. What does a database model need? What is in it? The following stages involve an examination of company objectives, company operations, and business rules for an OLTP database. The case study introduces concepts, such as analysis and design of relational database model for an online auction house company.

Project begins by analyzing and presenting the OLTP database model for the online auction house. Besides, the intention to establish what goes on operationally within the online auction house. What pieces of information make up the database model (static data), and what pieces of information are moving through the database model (transactional data)?

Oracle database system is used in the implementation part of the project.
# Contents

<table>
<thead>
<tr>
<th>Title</th>
<th>page no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Contents</td>
<td>ii</td>
</tr>
<tr>
<td>CHAPTER ONE: Introduction</td>
<td></td>
</tr>
<tr>
<td>1.1 Overview</td>
<td>1</td>
</tr>
<tr>
<td>1.2 An Online Auction House Overlook</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Characteristics of the Database approach</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Project objectives</td>
<td>3</td>
</tr>
<tr>
<td>1.5 Project outline</td>
<td>3</td>
</tr>
<tr>
<td>CHAPTER TWO: The OLTP Database Model</td>
<td></td>
</tr>
<tr>
<td>2.1 Auction House Categories</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Auction House Seller Listings</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Auction House Buyers And Bids</td>
<td>6</td>
</tr>
<tr>
<td>2.4 Auction House General Structure</td>
<td>7</td>
</tr>
<tr>
<td>2.5 Discovering Business Rules</td>
<td>7</td>
</tr>
<tr>
<td>2.6 Auction House Categories(ERDs)</td>
<td>8</td>
</tr>
<tr>
<td>2.7 Auction House Seller Listings(ERDs)</td>
<td>9</td>
</tr>
<tr>
<td>2.8 Auction House Buyers(ERDs)</td>
<td>11</td>
</tr>
<tr>
<td>2.9 Analyzing an OLTP Database Model</td>
<td>11</td>
</tr>
</tbody>
</table>
CHAPTER THREE: Creating and Refining the Database Application

3.1 Introduction ..................................................................................................12
3.2 Looking at Methods of Database Design.....................................................12
3.3 A Little More About Design................................. ...................................13
3.4 Creating Tables...............................................................................................14
3.5 Enforcing Table Relationships.......................................................................18
  3.5.1 Referential Integrity............................................................................. 18
3.5.2 Primary and Foreign Keys.......................................................................18
3.6 Project Functionality..........................................................................................20

CHAPTER FOUR: Conclusions and Future Work

4.1 Conclusions ...................................................................................................23
4.2 Future Work.....................................................................................................23
References............................................................................................................24
Chapter One
Introduction

1.1 overview

A number of characteristics distinguish the database approach from the traditional approach of programming with files. In traditional file processing, each user defines and implements the files needed for a specific application as part of programming the application. For example, one user, the grade reporting office, may keep a file on students and their grades. Programs to print a student’s transcript and to enter new grades into the file are implemented. A second user, the accounting office, may keep track of students’ fees and their payments.

Although both users are interested in data about students, each user maintains separate files—and programs to manipulate these files—because each requires some data not available from the other user’s files. This redundancy in defining and storing data results in wasted storage space and in redundant efforts to maintain common data up-to-date. In the database approach, a single repository of data is maintained that is defined once and then is accessed by various users.

1.2 An Online Auction House Overlook

An Auction House Does not produce anything but acts as an agent for a seller. A seller sells an item by offering it up for auction to any number of potential buyer. Buyers make bids until a specified time period has passed, at which point the auction winning bidder is decided. The auction house makes its revenue by charging the seller a commission for listing the item up for auction.

In this project, The following steps are performed

- The basics of analysis and design.
- The steps in the analysis process.
- Common problem areas and misconceptions associated with analysis.
- The value of following a paper trail.
- How to create a database model to cover objectives.
- How to refine a database model using business rules.
- How to apply everything learned so far with a comprehensive case study.
1.3 Characteristics of the Database Approach

The building of a database model can be divided up into distinct steps (as can any software development process).

1. Analysis
2. Design
3. Construction
4. Implementation

These steps can be loosely defined as follows:

Step 1: Analysis

Analyzing a situation or company is a process of initial fact-finding through interviews with end-users. If there are technical computer staff members on hand, with the added bonus of inside company operational knowledge, interview them as well.

Analysis is all about what a company does for a living? This equates to analyzing what are the tables in the database? And what are the most basic and essential relationships between those tables? Analysis defines general table structure. An auction Web site might contain a seller table and a bidder table. We must know what tables should generally contain in terms of information content. Should both the seller and bidder tables contain addresses of sellers and bidders respectively.

Step 2: Design

Design involves using what was discovered during analysis, and figuring out how those analyzed things can be catered for, with the software tools available. The analysis stage decided what should be created. The design stage applies more precision by deciding how tables should be created. This includes the tables, their fields, and datatypes. Most importantly, it includes how everything is linked together.

Referential integrity is a design process because it enforces relationships between tables. The initial establishment of inter-table relationships is an analysis process, not one of designs. The design stage introduces database modeling refinement processing, such as normalization.
Chapter One: Introduction

Step 3: Construction

In this stage, we build and test code. For a database model, we build scripts to create tables, referential integrity keys, and anything else such as stored procedures. In other words, with this step, we build scripts to create tables and execute them.

Step 4: Implementation

This is the final step in the process. Here we create a production database. In other words, it is at this stage of the process that we actually put it all into practice.

1.4 Project objectives

The first step in putting the database modeling process into practice is analysis. Discover and quantify what exactly it is that the company does to make ends meet. Begins with an examination of company objectives for the online auction house we examine company operations and business rules in two separate contexts: one for an OLTP database. Using the Internet, an online auction house is extremely likely to reach far more potential customers than an offline auction house.

This would make it possible for the auction house to market far less expensive products, in very much larger quantities. Obviously, commissions would be lower.

An online auction house can make good profitability, however, out of much higher quantities, all the while auctioning much cheaper items. So, an online auction house has a very large potential market of sellers and buyers. Technically, this makes the usage capacity of this database highly shareable (concurrent) between thousands of users, if not more. An OLTP database model structure is, therefore, desirable.

1.5 Project outline:

In chapter 1, we discuss the overview about the characteristics of database, overlook of an online auction house, the steps that performed in this project and the project objectives.

In chapter 2, we discuss the basics of analysis, a usable set of semi-formal steps, for the database modeling analysis process, some common problem areas and misconceptions that can arise, how talking, listening, and following a paper trail can be of immense value.

In chapter 3, we expand on the analysis process (simplistic in this chapter at best) and design with more detail provided for the OLTP database models, as presented analytically in chapter 2.

In chapter 4, we discuss the conclusions about the work at analysis (determining business rules), design and implementing stages, future work about how to develop the project.
Chapter Two

The OLTP Database Model

2.1 Auction House Categories

An online auction house acts as an agent for a seller. A seller sells an item by offering it up for auction to any number of potential buyers. Buyers make bids until a specified time period has passed, at which point the auction winning bidder is decided. The auction house makes its revenue by charging the seller a commission for listing the item up for auction. Examine the operational aspects for an online auction house database model.

There are a number of category layers: The primary category layer divides all listed items into primary categories, such as musical instruments, books, antiques, collectibles, toys, hobbies. There are others that will be detailed at a later stage.

The secondary category layer is effectively a subcategory within each of the primary categories. For example, the category for musical instruments can contain secondary category items, such as brass instruments, woodwind instruments, and stringed instruments.

Figure 2-1 shows a simple diagram of the structure of the category information for our online auction house. Each secondary has a primary category parent.

![Diagram of category structure](image)

Figure 2-1: Static categories information placed as static parts of the database model.
2.2 Auction House Seller Listings

Auction listings are essentially an item a seller is selling, listed on the auction site for sale.

- Each auction listing contains a link through to details about the seller (the person or organization selling the item), such as the seller’s name and address, the starting price of the item, and shipping details (shipping is required for an online auction house to determine who and where an item should be sent to, upon auction listing completion.

- An auction listing must have a seller, but not necessarily a final buyer. If there are no bids placed on the item, no buyer buys the auctioned item—there is no buyer.

- Auction listings have links to other subsets, which are one-to-many master-detail sets of information:
  - Each auction can have a link to a history about the seller. Is the seller a reputable seller? This section consists of reviews of how past buyers have felt about dealing with a particular seller.
  - Each auction item can have a link to a history of bids on a particular auction item. Both sellers and buyers could then examine all past bids made on an auctioned item, over the life of each auction listing.

Figure 2-2 shows the relationships between a seller and their auction listings, and their history of past dealings with buyers. Sellers can have more than one listing, not only over a period of time, but can list more than one item at the same time. There are also special types of auctions where a single listing can list multiple items for sale to multiple possible bidders, depending on how many items a bidder wishes to buy.

![Figure 2-2: Sellers, listings, and seller history (dynamic information moving through the database model).](image-url)
2.3 Auction House Buyers And Bids

Buyers have various items of information:

- Buyers should have bidding records so that sellers can verify they are dealing with a reputable buyer. It is prudent for sellers to avoid accepting bids on items for buyers who have, for example, placed bids in the past without making payment.
- The auction site needs to store buyer information such as names, addresses, credit card information, and credit worthiness.

Figure 2-3 shows the relationships between a buyer, bids made on existing listings, and a history of past dealings with buyers. Any listing can have more than one bid made on it during the life of the listing. When the auction is complete and there is an auction winner, no more bids are allowed. A listing may have no bids at all, at which stage it can be listed again.

![Diagram showing relationships between buyers, listings, bids, and buyer history](image)

Notice how the BIDS table is connected to both the LISTING and BUYER tables. This is the only table in this database structure that is connected to more than one table and not as part of a hierarchy. Category tables are part of a hierarchy.

Once again, notice the dual links for the BID table (now called BID for technical accuracy because each record represents a single bid), to both the LISTING and the BUYER tables. This double link represents a many-to-many relationship. The BID table is in actuality a resolution of the many-to-many relationship between the LISTING and BUYER tables. In other words, each buyer can place many bids on a single item listed for auction; however, each auction listed item can also have many buyers making those many bids.
2.4 Auction House General Structure

Figure 2-4 shows an overall picture of the entire operational structure for the company, as discovered so far.

![Diagram of Auction House General Structure](image)

One of the main points to remember in any software development project is that the steps in methodologies are iterative, and even the sequence in which the steps are executed is not necessarily strictly applied. For example, in the middle of analyzing and defining in the business rule stage, you can always go back and rework the company operations section, as shown in Figure 2-5.

2.5 Discovering Business Rules

So, begin an application of business rules by first examining what a business rule is. In short, business rules for a business are a set of functional rules that can be used to describe a business and how it operates, essentially a semi-mathematical interpretation of how a business functions. Business rules are essentially a more precise interpretation of a company’s operational activities, as established for the online auction house in the previous section.
The business rules of a company, when applied to a database model, become that database model. The business rules are the tables, relationships between those tables, and even the constraint rules (in addition to referential integrity constraints) in a database. In fact, the application of normalization to a set of initial company data tables is a more and more complex application of business rules to that company data environment. Normalization is the application of business rules.

![Diagram](image)

**Figure 2-5: Iterative steps in the analysis methodology.**

As already stated, business rules are at the heart of the analysis stage, describing what has been analyzed and what must be created to design the database model. The overall aim of analysis is merely to define rather than specify with precision; therefore, analysis does not describe how many fields should be used for an address, for example, or what data types those fields should be. Analysis simply determines that an address field actually exists, and obviously which table or tables address information is required within.

At this stage, you can take the operations of the online auction house, established previously in Figure 2-1 to Figure 2-4, and create Entity Relationship Diagrams (ERDs) for those structures.

### 2.6 Auction House Categories (ERDs)

Figure 2-6 caters effectively to the category hierarchical structure, and the link to the table containing auction listings. The LISTING table has all the details for the master side of a master—detail table that depicts table design, including a description, listing table number reference (LISTING#), dates, prices, bids, and winning bidder details.
Figure 2-6: An ERD version of the category hierarchical structure of Figure 2-1.

2.7 Auction House Seller Listings (ERDs)

Figure 2-7 shows basic table structure as related to auction listings—the sellers of the listings (the person or organization selling something at auction). All that is needed in the analysis stage is basic tables and relationships, including a preliminary field structure for each table.
Figure 2-7: Adding seller information to the category structure in Figure 2-6.

In Figure 2-7, the seller and seller history information has been added with various field items to represent details of both. Seller history information would include details such as who the buyer was, what was purchased (the auction item listing number), and what the buyer had to say about the seller (comments), among other details. Historical information allows the auction house to give a popularity rating to a seller, and also allows any future buyers to assess the reputation of a buyer that the sellers may or may not wish to deal with.
2.8 Auction House Buyers (ERDs)

Figure 2-8 adds the buyers to the table structure established so far. A buyer details table and a buyer history table are added. Buyer history is the same as seller history field information.

2.9 Analyzing an OLTP Database Model

Here’s a simplistic approach:

1. Identify the operations of the company.
2. Draw up a picture of basic tables.
3. Establish simple relationships.
4. Create basic fields within each table.
Chapter Three

Creating and Refining the Database Application

3.1 Introduction

Analysis is all about what needs to done. Design does it! This chapter builds on and expands on the basic analytical process and structure discovered during the case study approach in Chapter two, which covered analysis. Analysis is the process of discovering what needs to be done. Analysis is all about the operations of a company, what it does to make a living. Design is all about how to implement that which was analyzed into a useful database model.

This chapter passes from the analysis phase into the design phase by discovering through a case study example how to build and relate tables in a relational database model. It explains how database models are created and designed. This chapter outlines how to begin the implementation of what was analyzed (discovered) in Chapter two. In short, implementation is the process of creating tables, and sticking those tables together with appropriate relationships.

The main stages are:
- Database model design
- Creating tables
- Enforcing inter-table relationships and referential integrity
- Normalization without going too far

3.2 Looking at Methods of Database Design

- Requirements analysis—Collect information about the nature of the data, features required, and any specialized needs such as expected output responses. This step covers what is needed, so simply analyze it and write it down. Talk to the customer and company employees to get a better idea of exactly what they need.
Chapter Three: Creating and Refining the Database Application

- **Conceptual design**—This is where you get to use the fancy graphical tools and draw the pretty pictures—Entity Relationship Diagrams (ERDs). This step includes creation of tables, fields within those tables, and relationships between the tables.

- **Logical design**—Create database language commands to generate table definitions. Some tools used for creating ERDs allow generation of data definition language (DDL) scripting; however, they are likely to generate generic scripts.

- **Data definition language (DDL)** is made of the commands used to change metadata in a database, such as creating tables, changing tables, and dropping tables.

- **Physical design**—Adjust database language commands to alter the database model for the underlying physical attributes of tables. For example, you might want to store large binary objects in separate, underlying files to that of standard relational record-field data.

- **Tuning phase**—This step includes items such as appropriate indexing, further normalization, or even de-normalization, security features, and anything else not covered by the previous steps.

These separate steps are interchangeable, repeatable, iterative, and really anything-able, according to various different approaches used for different database engines and different database designer personal preferences. Some designers may even put some of these steps into single steps and divide others up into more detailed sets of subset steps. In other words, this is all open to interpretation. The only thing I do insist that should be universal is that you draw the ERDs and build tables well before you build metadata table creation code, placing visual design prior to physical implementation.

3.3 A Little More About Design

So, the design stage is the next stage following the analysis stage. Design is a process of figuring out how to implement what was discovered during the analysis stage. As described in Chapter two, analysis is about what needs to be done. Design is about how it
should be done. The design stage deals with the following aspects of database model creation:

- More precise tables, including practical application of normalization.
- Establishment of primary and foreign key fields.
- Enforcement of referential integrity by establishing and quantifying precise relationships between tables, using primary and foreign key fields.
- Precise field definitions, structure, and their respective data types (again advanced design).

3.4 Creating Tables

In Chapter 2, tables were created on an analytical level, creating basic pictures. Following the basic pictures, simple ERDs were constructed. In this section, basic commands are used to create the initial simple tables, as shown in the analytical process of Chapter 2. The idea is to retain the step-by-step instruction of each concept layer, in the database modeling design process, for OLTP database model. This model is created for the online auction house case study database models.

Figure 3-2: Business rules OLTP online auction house database model.
Chapter Three: Creating and Refining the Database Application

The easiest way to create tables in a database model is to create them in a top-down fashion, from static to dynamic tables, gradually introducing more and more dependent detail. In others words, information that does not change very often is created first. Information changing frequently is created after static tables. Technically speaking, it’s all about catering to dependencies first. The first tables created are those with no dependencies. Subsequent tables are created when tables depended on have already been created. Begin by creating the two category static tables. The SECONDARY table depends on the PRIMARY table, therefore we need to create PRIMARY, then SECONDARY tables:

```
CREATE PRIMARY TABLE

CREATE TABLE "SARA"."PRIMARY"
  (  "P_ID" NUMBER(2) NOT NULL,
     "PNAME" VARCHAR2(50) NOT NULL,
     CONSTRAINT "PRIMARY_PK" PRIMARY KEY("P_ID")  )
TABLESPACE "USERS"
```

```
CREATE SECONDARY TABLE

CREATE TABLE "SARA"."SECONDARY"
  (  "SC_ID" NUMBER(2) NOT NULL,
     "P_ID" NUMBER(2),
     "SCNAME" VARCHAR2(50) NOT NULL,
     CONSTRAINT "SECONDARY_PK" PRIMARY KEY("SC_ID"),
     CONSTRAINT "SECONDARY_FK" FOREIGN KEY("P_ID")
     REFERENCES "SARA"."PRIMARY"("P_ID")  )
TABLESPACE "USERS"
```

The SELLER and BUYER tables are static. According to the ERD shown in Figure 3-2 they are not dependencies. So we can create the SELLER and BUYER tables next:

```
CREATE SELLER TABLE

CREATE TABLE "SARA"."SELLER"
  (  "S_ID" NUMBER(3) NOT NULL,
     "SNAME" VARCHAR2(40) NOT NULL,
     "POP_RATE" NUMBER(3),
     "JOIN_DATE" DATE,
     "ADD" VARCHAR2(30),
     "PAY_METH" VARCHAR2(30),
     CONSTRAINT "SELLER_PK" PRIMARY KEY("S_ID")  )
TABLESPACE "USERS"
```

```
CREATE BYUER TABLE

CREATE TABLE "SARA"."BUYER"
  (  "B_ID" NUMBER(3) NOT NULL,
     "BNAME" VARCHAR2(40) NOT NULL,
     "POP_RATE" NUMBER(3),
     "JOIN_DATE" DATE,
     "ADD" VARCHAR2(30),
     CONSTRAINT "BUYER_PK" PRIMARY KEY("B_ID")  )
```
Note how the SELLER_HISTORY table does not have a SELLER field, because this is implied by the direct parent relationship to the SELLER table. The same applies to the BUYER_HISTORY table, containing the SELLER field only.

```sql
{ CREATE SELLER HISTORY TABLE

CREATE TABLE "SARA"."SELLER_HIS"
( "S_H_ID" NUMBER(2) NOT NULL,
  "S_ID" NUMBER(2) NOT NULL,
  "B_ID" NUMBER(2),
  "COM_DATE" DATE,
  "LIST#" NUMBER(2),
  "COMMENTS" VARCHAR2(255),
  CONSTRAINT "SELLER_HIS_PK" PRIMARY KEY("S_H_ID"),
  CONSTRAINT "SELLER_HIS_FK1" FOREIGN KEY("S_ID")
    REFERENCES "SARA"."SELLER"("S_ID"),
  CONSTRAINT "SELLER_HIS_FK2" FOREIGN KEY("B_ID")
    REFERENCES "SARA"."BUYER"("B_ID")
)
TABLESPACE "USERS"
}

{ CREATE BUYER HISTORY TABLE

CREATE TABLE "SARA"."BUYER_HIS"
( "B_H_ID" NUMBER(2) NOT NULL,
  "B_ID" NUMBER(2) NOT NULL,
  "S_ID" NUMBER(2),
  "COM_DATE" DATE,
  "LIST#" NUMBER(2),
  "COMMENTS" VARCHAR2(255),
  CONSTRAINT "BUYER_HIS_PK" PRIMARY KEY("B_H_ID"),
  CONSTRAINT "BUYER_HIS_FK1" FOREIGN KEY("B_ID")
    REFERENCES "SARA"."BUYER"("B_ID"),
  CONSTRAINT "BUYER_HIS_FK2" FOREIGN KEY("S_ID")
    REFERENCES "SARA"."SELLER"("S_ID")
)
TABLESPACE "USERS"
}

Next, create the LISTING table:

{ CREATE LISTING TABLE

CREATE TABLE "SARA"."LISTING"
( "LIST#" NUMBER(2) NOT NULL,
  "SC_ID" NUMBER(2) NOT NULL,
  "S_ID" NUMBER(2) NOT NULL,
  "B_ID" NUMBER(2),
  "DESCR" VARCHAR2(255),
  "IMAGE" LONG RAW,
  "ST_DATE" DATE,
  "LIST_DAYS" NUMBER(2),
  "CURRENCY" VARCHAR2(255),
  "ST_PRI" NUMBER(10) NOT NULL,
  "WIN_PRI" NUMBER(10),
  CONSTRAINT "LISTING_PK" PRIMARY KEY("LIST#"),
  CONSTRAINT "LISTING_FK1" FOREIGN KEY("SC_ID")
    REFERENCES "SARA"."SECONDARY"("SC_ID"),
  CONSTRAINT "LISTING_FK2" FOREIGN KEY("S_ID")
    REFERENCES "SARA"."SELLER"("S_ID"),
  CONSTRAINT "LISTING_FK3" FOREIGN KEY("B_ID")
)
TABLESPACE "USERS"
}
```
REFERENCES "SARA"."BUYER"("B_ID")
TABLESPACE "USERS"

Lastly, create the BID table (the BID table is dependent on the LISTING table):

```sql
CREATE TABLE "SARA"."BID"
(   "B_ID" NUMBER(2),
    "BIDDER" VARCHAR2(50),
    "SC_ID" NUMBER(2),
    "PRICE" NUMBER(10),
    CONSTRAINT "BID_FK1" FOREIGN KEY("B_ID")
        REFERENCES "SARA"."BUYER"("B_ID"),
    CONSTRAINT "BID_FK2" FOREIGN KEY("SC_ID")
        REFERENCES "SARA"."SECONDARY"("SC_ID")
) TABLESPACE "USERS"
```

Now, we must create new user and grant him the permissions to create tables and confirm relationship among tables.

```sql
CREATE USER "SARA"  PROFILE "DEFAULT"
    IDENTIFIED BY "sara" DEFAULT TABLESPACE "USERS"
    ACCOUNT UNLOCK;
    GRANT ADMINISTER DATABASE TRIGGER TO "SARA"
    BEGIN
    BEGIN
        dbms_resource_manager_privs.grant_system_privilege(privilege_name=>'ADMINISTER_RESOURCE_MANAGER', grantee_name=>'SARA', admin_option=>FALSE);
    END;
    GRANT ALTER ANY TABLE TO "SARA"
    GRANT ALTER DATABASE TO "SARA"
    GRANT BACKUP ANY TABLE TO "SARA"
    GRANT COMMENT ANY TABLE TO "SARA"
    GRANT CREATE ANY PROCEDURE TO "SARA"
    GRANT CREATE ANY TABLE TO "SARA"
    GRANT CREATE ANY TRIGGER TO "SARA"
    GRANT SELECT ANY TABLE TO "SARA"
    GRANT SYSDBA TO "SARA"
    GRANT SYSOPER TO "SARA"
    GRANT "AQ_ADMINISTRATOR_ROLE" TO "SARA";
    GRANT "AQ_USER_ROLE" TO "SARA";
    GRANT "CONNECT" TO "SARA";
    GRANT "DBA" TO "SARA";
    GRANT "EXP_FULL_DATABASE" TO "SARA";
    GRANT "IMP_FULL_DATABASE" TO "SARA";
    GRANT "RESOURCE" TO "SARA";
    END;
```
Chapter Three: Creating and Refining the Database Application

3.5 Enforcing Table Relationships

Referential integrity maintains and enforces the data integrity of relationships between tables. In other words, referential integrity ensures that where a child table record exists, the parent table record exists as well.

3.5.1 Referential Integrity

In Figure 3-1, you cannot delete a SELLER record without deleting all the seller’s listings first. If the seller is deleted, their listings become orphaned records. An orphaned record is term applied to a record not findable within the logical table structure of a database model. Essentially, the seller’s name and address details are stored in the SELLER table and not in the LISTING table. If the seller record was deleted, any of their listings are useless because you don’t know who is selling it. Similarly, if buyer information with winning bids were deleted, the seller wouldn’t know who to mail it to. Referential integrity, through the use of primary and foreign keys, acts to ensure that the following activities are prohibited:

- INSERT check—A child record cannot be added to a child table unless the parent record exists in the parent table.
- UPDATE check—Parent and child table records cannot have their linking key field values changed, unless both are changed simultaneously (“simultaneously” implies within the same transaction).
- DELETE check—A parent table record cannot be deleted when a child table record exists, unless all related child table records are deleted first.
- DELETE CASCADE check—A parent table record can be deleted if all child records are deleted first. This is known as a cascading delete. Cascade deletion is rarely implemented because it can result in serious data loss (usually through operator or programmer error).

Additionally, cascade deletions can cause serious locking conflicts because large numbers of child records could have to be deleted, when deleting a parent record. This type of locking problem can occur when a parent record has many records in child tables, through a multiple level table structure.

3.5.2 Primary and Foreign Keys

Primary and foreign keys are used to establish referential integrity relationships between parent and child tables. The parent table contains the primary key, and the child table the foreign key. The term primary implies most significant field for a table, and thus uniquely identifying. Each seller record would have a unique seller name (the name of the
person or company selling the item at auction). Two sellers can’t have the same name, leading to the obvious silly result. How would you differentiate between two different sellers? Impossible. The term foreign implies a key that is foreign to a child table, whose uniquely identifying value lies in another table (the parent table containing the primary key).

Now demonstrate implementation of primary and foreign keys by re-creating the OLTP database model tables, as shown in Figure 3-1. The ERD in Figure 3-1 has been changed to the ERD shown in Figure 3-3.

Figure 3-3: Defining primary and foreign keys for the OLTP online auction house database model.

The database model is the backbone of any application that uses data of any kind. That data is most likely stored in some kind of database. That database is likely to be a relational database of one form or another.

Better designed database models tend to lend themselves to clearer and easier construction of SQL code queries. The ease of construction of, and the ultimate performance of queries, depends largely on the soundness of the underlying database model.
Chapter Three: Creating and Refining the Database Application

The better the database model design, the better queries are produced. The efficiency of queries comes down to how many tables are joined in a single query.

3.6 Project Functionality

In this section some of the project functionalities are presented. First an algorithm of an online auction is listed in which the system choose the buyer who gives the maximum price.

Algorithm of an online auction
Input: goods and price for each
Output: adding records to the bidder table
Variable: listing, bidder
Method
begin
Step1: seller will put his article and the start price
Step2: one or more buyers will give their bids
Step3: system will choose the buyer with the maximum bid
end

The following figures illustrate the functions outlined along this project

Figure 3-4: The main application frontal
Chapter Three: Creating and Refining the Database Application

Figure 3-5: The buyers frontal

Figure 3-6: The customers frontal
Figure 3-7: The listings frontal

Figure 3-8: The listings report
Chapter four

Conclusions and Suggestions

4.1 Conclusions

The best database models are produced by paying attention to detail in the analysis stage. It is important to understand exactly what is needed before jumping in and “just getting to it.”

Making changes to a database model used in a production system can be extremely problematic. This is because applications are usually dependent on the database model. The reason is that the database model forms the basis of all database-driven applications, quite often for an entire company. Business rules—This is the heart of the analysis stage, describing what has been analyzed and what needs to be created to design the database model. What tables are needed and what are the basic relationships between those tables? Excessive application of normalization removes potential for error in data (referential integrity violations).

4.2 Suggestions

Also, it is valuable to the auction house (and perhaps many of its selling clientele) to understand trends. Trends and forecasting can be established using forecasts and projects into archived data sets. Archived data can be stored conveniently and efficiently using a specialized data warehouse database.

- Denormalization in the design stage (the sooner the better).
- Alternate and extra indexing in addition to that of referential integrity, primary and foreign keys; however, alternate indexing is more advanced (detailed) design, and is discussed in Chapter 3.
- Advanced database structures, such as materialized views, and some specialized types of indexing. Similar to alternate indexing, this is also more advanced (detailed) design, and is discussed in Chapter 3.
References: