



Zagazig University
Faculty of computer & information
Computer Science (CS).

SCHEDULES

Supervisor:

Prof. Mohamed Abdelbast

2021

Team Members:

- 1- Mustafa Mohamed Mustafa Hamed**
- 2- Mohamed Elsayed Mohamed Mesallam Sayed Ahmed**
- 3- Sara Salem Abdelwahd Salem**
- 4- Nehal Mahmoud Abdelkhalek**
- 5- Mayar Ragab Elsayed Abdelrazik**
- 6- Hebaallah Ashraf Elshikh**
- 7- Zaynab Younis Younis Yousif Elshihk**

Supervisor

**Prof. Mohamed Abdelbast
Faculty of Computer and informatics
Zagazig University**

Acknowledgment

We would like to express our sincere gratitude and respect to our faculty staff members and everyone who helped us during the graduation project, starting with endless thanks for our supervisor **Prof. Mohamed Abdelbast** who didn't keep any effort in encouraging us to do a great job, providing our group with valuable information and advices to be better each time. Thanks for the continuous support and kind communication which had a great effect regarding to feel interesting about what we are working on, and who directed us to new color of science, drawing and planning most of all our steps providing us good resources and references for finishing that work.

Abstract

With the recent and fast developments in technology, The computer is the most used device of the modern era and you cannot do anything without the computer even in your work, you need the computer greatly to finish the tasks that you are assigned to in your work, and when we define the computer desktop applications offer many advanced services, so the number of computer users increases day by day. User can install and run many desktop applications that make life easier and enjoyable.

In our world there is more development and we must adapt to this development so we got the idea we should use this technology to give a hand for our college and university, After thinking and take advices from teaching stuff we decided to make our schedules system just with one click you can create or modify the whole collage schedule, This will save time and produce a lot of options, to help those who create the schedule and make them take a breath from being feel they must take another look on the schedule, And we hope that our system could help my college and university to save time and efforts to just create the year plans.

Table of content

Acknowledgment.....	3
Abstract.....	4
Chapter 1: Introduction	
1.1 Problem Statement	
1.2 Project Objectives	
1.3 Importance To People	
1.4 The Overall Project	

Chapter One

Introduction

1.1 Problem Statement

When someone who need to sets the schedule, faces some problems:

- Number of academic years that may be divided into double the number, because of the divided of the one academic year. This results in the creation of two lecture schedules for each academic year.
- Number of sections which can divide one academic year into twenty sections or more.
- Working hours for university Professors and Teaching Assistants.
- Holidays for each academic year.
- Period Start and Finish time for each lecture and section.
- Numbers of halls and laboratories.

1.2 Project Objectives

Our application is built for helping collage to solve the schedule problem and finding the fastest solution to the problem by using desktop application, push you some questions and by answering and click on create table will display schedule.

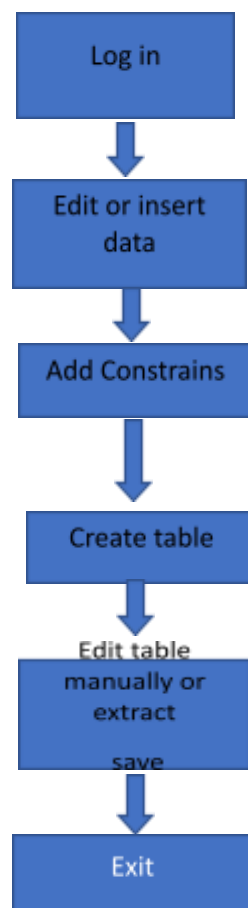
- When user wants to enter in the application, the application asks them to gather some information about user: username and password.
- When user registered in the application, He has many choices that allows to edit, delete or add any data.

- Application allows to create a schedule manually or automatically, by determines that.
- This application solve all problems that faces college in set schedule.
-

1.3 Importance to College

- This most important features in this application is that you can get a university schedule without charge (no money paid) in less time.
- Save time and efforts.
- Reduce the conflict that may case by human.

1.4 The Overall Project



1.5 Competitive Advantages

After search for application that can help university in setting tables we can't find it. So we looking forward to develop this application to performs better.

1.6 Documentation Guideline

This is The Official Documentation for our graduation project. Project name Schedule...This project provide university with modern way to get a schedule easy and fast and also decrease efforts and time.

This Documentation Divided into chapters and Every Chapter Talking about part or section of the Project;

Chapter one that we are here is Introduction where Talking about the Problem that give us the idea to create this application. The introduction chapter also contains over view for the project structure.

Chapter Two that Provide General Over view About the tools and Services That Being Used in This Application and Simple Over view about each one of this tools and also provide the feature and updates that give the application value and make it differ than any other application talking about Competitive Advantages .

Chapter Three That Provide overview about the backend subsystem that being used in the application that divided into two parts the first one is

the Firebase Technology for The General Project Database that represented in Registration and users Contacts and some other related Data.

Chapter Four that contain the implementation for the Project and how the user will work with the project ...what is the screens contents.

Chapter five in the Project represent the analysis for the project and what is the use case for the project, also give the Data Flow Diagrams for the Project, and provide full Details for Illustrate the Data Flow and the Project Work.

Chapter Six that being in Title Conclusions and give an Overview about the Team Vision and what the Future Updates and the Next Step for the Team Work.

The Final Part in the Project provide references for all recourse that we depend on the project and that we use in our Learning process or Testing.

Chapter Two

Related Background

2.1 GUI And Application Language

2.1.1 GUI

- **The graphical user interface (GUI)** is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs), which require commands to be typed on a computer keyboard.
 - The actions in a GUI are usually performed through direct manipulation of the graphical elements. Beyond computers, GUIs are used in many handheld mobile devices such as MP3 players, portable media players, gaming devices, smartphones and smaller household, office and industrial controls. The term GUI tends not to be applied to other lower-display resolution types of interfaces, such as video games (where head-up display (HUD) is preferred), or not including flat screens, like volumetric displays because the term is restricted to the scope of two-dimensional display screens able to describe generic information, in the tradition of the computer science research at the Xerox Palo Alto Research Center.

- A GUI uses a combination of technologies and devices to provide a platform that users can interact with, for the tasks of gathering and producing information.
- A series of elements conforming a visual language have evolved to represent information stored in computers. This makes it easier for people with few computer skills to work with and use computer software. The most common combination of such elements in GUIs is the windows, icons, menus, pointer (WIMP) paradigm, especially in personal computers.
- The WIMP style of interaction uses a virtual input device to represent the position of a pointing device's interface, most often a mouse, and presents information organized in windows and represented with icons. Available commands are compiled together in menus, and actions are performed making gestures with the pointing device. A window manager facilitates the interactions between windows, applications, and the windowing system. The windowing system handles

hardware devices such as pointing devices, graphics hardware, and positioning of the pointer.

- In personal computers, all these elements are modeled through a desktop metaphor to produce a simulation called a desktop environment in which the display represents a desktop, on which documents and folders of documents can be placed.

2.1.2 JAVA

- **Java** is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages.

- Swing application Main article: [Swing \(Java\)](#)

Swing is a graphical user interface library for the Java SE platform. It is possible to specify a different look and feel through the pluggable look and feel system of Swing. Clones of Windows, GTK+, and Motif are supplied by Sun. Apple also provides an Aqua look and feel for macOS. Where prior implementations of these looks and feels may have been considered lacking, Swing in Java SE 6 addresses this problem by using more native GUI widget drawing routines of the underlying platforms.

2.1.3 JAVA Swing

- **Swing** is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.
 - Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check

boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

- Unlike AWT components, Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and therefore are platform-independent.
- Swing is a platform-independent, "model–view–controller" GUI framework for Java, which follows a single-threaded programming model. Additionally, this framework provides a layer of abstraction between the code structure and graphic presentation of a Swing-based GUI.

● Relationship to AWT

AWT and Swing class hierarchy

Since early versions of Java, a portion of the Abstract Window Toolkit (AWT) has provided platform-independent APIs for user interface components. In AWT, each component is rendered and controlled by a native peer component specific to the underlying windowing system.

By contrast, Swing components are often described as lightweight because they do not require allocation of native

resources in the operating system's windowing toolkit. The AWT components are referred to as heavyweight components. Much of the Swing API is generally a complementary extension of the AWT rather than a direct replacement. In fact, every Swing lightweight interface ultimately exists within an AWT heavyweight component because all of the top-level components in Swing (JApplet, JDialog, JFrame, and JWindow) extend an AWT top-level container. Prior to Java 6 Update 10, the use of both lightweight and heavyweight components within the same window was generally discouraged due to Z-order incompatibilities. However, later versions of Java have fixed these issues, and both Swing and AWT components can now be used in one GUI without Z-order issues.

2.1.4 PL/SQL

- **PL/SQL (Procedural Language for SQL)** is Oracle Corporation's procedural extension for SQL and the Oracle relational database.
- **PL/SQL** includes procedural language elements such as conditions and loops. It allows declaration of constants and variables, procedures and functions, types and variables of those types, and triggers. It can handle exceptions (run-time errors). Arrays are supported involving the use of PL/SQL collections. Implementations from version 8 of Oracle

Database onwards have included features associated with object-orientation. One can create PL/SQL units such as procedures, functions, packages, types, and triggers, which are stored in the database for reuse by applications that use any of the Oracle Database programmatic interfaces.

- **PL/SQL anonymous block**

The basic unit of a PL/SQL source program is the block, which groups together related declarations and statements. A PL/SQL block is defined by the keywords DECLARE, BEGIN, EXCEPTION, and END. These keywords divide the block into a declarative part, an executable part, and an exception-handling part. The declaration section is optional and may be used to define and initialize constants and variables. If a variable is not initialized then it defaults to NULL value. The optional exception-handling part is used to handle run time errors. Only the executable part is required. A block can have a label

2.2 Software Tools

2.2.1 Introduction To Oracle:

- **Oracle Database** (commonly referred to as Oracle DBMS or simply as Oracle) is a multi-model database management system produced and marketed by Oracle Corporation.

- It is a database commonly used for running online transaction processing (OLTP), data warehousing (DW) and mixed (OLTP & DW) database workloads. Oracle Database is available by several service providers on-prem, on-cloud, or as hybrid cloud installation. It may be run on third party servers as well as on Oracle hardware (Exadata on-prem, on Oracle Cloud or at Cloud at Customer)

- **Why Oracle** An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

- Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular

storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

- The database has logical structures and physical structures.

Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

- **Overview of Scalability and Performance Features**

- Oracle includes several software mechanisms to fulfill the following important requirements of an information management system:
- Data concurrency of a multiuser system must be maximized.
- Data must be read and modified in a consistent fashion. The data a user is viewing or changing is not changed (by other users) until the user is finished with the data.
- High performance is required for maximum productivity from the many users of the database system.

2.2.3 Introduction To Toad:

- **Toad** is a database management toolset from Quest Software for managing relational and non-relational databases using SQL aimed at database developers, database administrators, and data analysts. The Toad toolset runs against Oracle, SQL Server, IBM DB2 (LUW & z/OS), SAP and MySQL. A Toad product for data preparation supports many data platforms.

- **Toad Features**

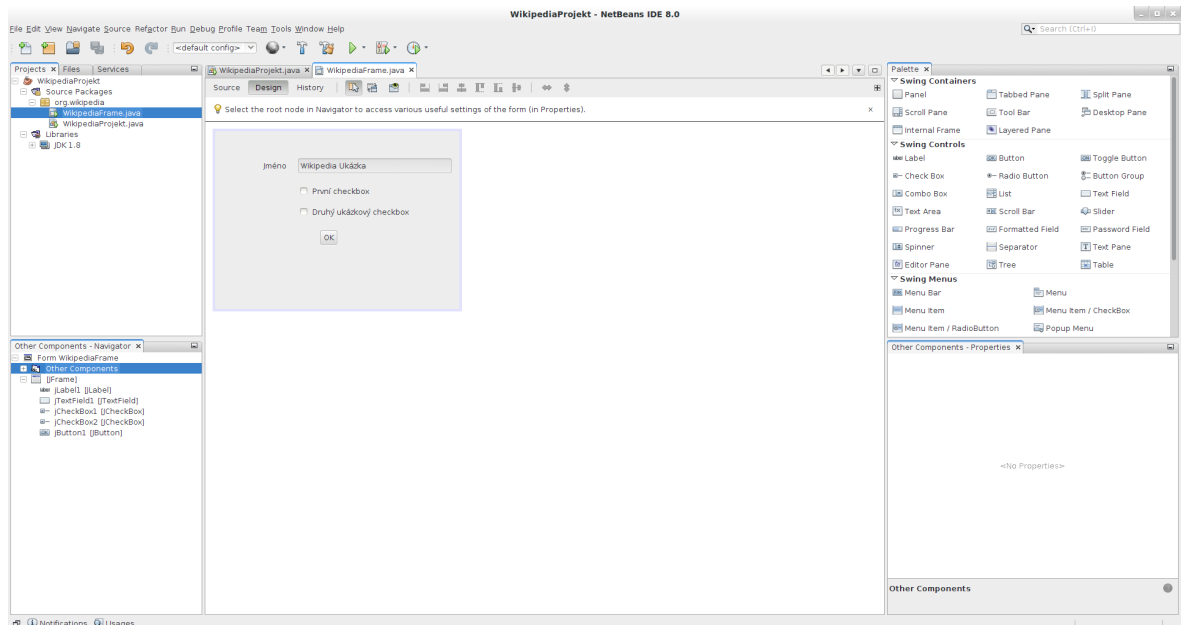
- Connection Manager - Allow users to connect natively to the vendor's database whether on-premise or DBaaS.
- Browser - Allow users to browse all the different database/schema objects and their properties effective management.
- Editor - A way to create and maintain scripts and database code with debugging and integration with source control.
- Unit Testing (Oracle) - Ensures code is functionally tested before it is released into production.
- Static code review (Oracle) - Ensures code meets required quality level using a rules-based system.

- SQL Optimization - Provides developers with a way to tune and optimize SQL statements and database code without relying on a DBA. Advanced optimization enables DBAs to tune SQL effectively in production.

2.2.2 Introduction To NetBeans IDE:

- **NetBeans IDE** is an open-source integrated development environment. NetBeans IDE supports development of all Java application types (Java SE (including JavaFX), Java ME, web, EJB and mobile applications) out of the box. Among other features are an Ant-based project system, Maven support, refactorings, version control (supporting CVS, Subversion, Git, Mercurial and ClearCase).
- **GUI design tool** NetBeans GUI Builder
Formerly known as project Matisse, the GUI design-tool enables developers to prototype and design Swing GUIs by dragging and positioning GUI components

The GUI builder has built-in support for JSR 295 (Beans Binding technology), but the support for JSR 296 (Swing Application Framework) was removed in 7.1



2.2.4 Introduction To Visual Paradigm:

- **Visual Paradigm (VP-UML)** is a UML CASE Tool supporting UML 2, SysML and Business Process Modeling Notation (BPMN) from the Object Management Group (OMG). In addition to modeling support, it provides report generation and code engineering capabilities including code generation. It can reverse engineer diagrams from code, and provide round-trip engineering for various programming languages.
- Visual Paradigm supports both Entity Relationship Diagrams (ERD) and Object Relational Mapping Diagrams (ORMD). ERD is used to model the relational database. ORMD is one of the tools to show

the mapping between class from object-oriented world and entity in relational database world.

● **Visual Paradigm Features**

- Visual Modeling
- Enterprise Architecture
- Business Analysis & Design
- Project Management
- Agile & Scrum Development
- Online Diagrams

Chapter Three

BackEnd

Welcome to Toad



Toad for Oracle provides an intuitive and efficient way for database professionals of all skill and experience levels to perform their jobs with an overall improvement in workflow effectiveness and productivity. With Toad for Oracle you can:

- Understand your database environment through visual representations
- Meet deadlines easily through automation and smooth workflows
- Perform essential development and administration tasks from a single tool
- Deploy high-quality applications that meet user requirements;

perform predictably and reliably in production

- Validate database code to ensure the best-possible performance and adherence to best- practice standards
- Manage and share projects, templates, scripts, and more with ease

The Toad for Oracle solutions are built for you, by you. Over 10 years of development and feedback from various communities like Toad World have made it the most powerful and functional tool available. With an installed-base of over two million, Toad for Oracle continues to be the “de-facto” standard tool for database development and administration.

Note: If you are using an older version of Toad, you can save *hours* with features introduced in later versions, notably Toad 12, such as Jump search, the Private Script Repository, Compare Multiple Schemas, the new Team Coding Dashboard, access to Toad World User Forums and how-to videos directly within Toad, Workspaces, Code Analysis(formerly Code Xpert), and Query Builder.

[5.1.1 To create a Workspace](#)

» In the Workspace toolbar, click  to name and save your Workspace.

Your open windows and connections are saved and will be reestablished the next time you open this Workspace.

These components are saved:


Schema Browser - The currently active database objects (type and name)

Editor - The tab contents, the number of tabs, the last active tab,

each tab's line and caret position, and the split mode of the Editor

Other windows - All other MDI style windows are managed with Workspaces. Non-MDI windows, that is, docked windows such as the Project Manager, are retained across Workspaces since they likely represent an overall working desktop state you may need across Workspaces.

5.1.2 To save changes to your Workspace configuration

- » Click  in the Workspace toolbar to save changes to your Workspace configuration (newly opened windows and connections).

You can customize schema drop-downs by creating a list of favorites, hiding schemas, setting the default schema for connections, and other options. Changes apply to all windows with the schema drop-down, such as the Editor and Schema Browser.

5.1.3 To set a default schema

- » Right-click the schema in the schema drop-down and select **Set <schema name> to Default Schema**.

5.1.4 To customize schema drop-downs

1. Right-click the schema drop-down and select **Customize**.
2. Select schemas to categorize and click the > button.
3. To hide schemas, select *Hidden Schemas* in the **Category** field for the schema.
4. To create a new category, enter the category name in the **Category**

field for the schema. The new name becomes available in the Category drop-down.

5. To change when the schema is categorized, select the When to Categorize field for the schema and click.

You can customize Toad's default menus and toolbars, and you can create new ones with custom options. This lets you arrange Toad to best reflect how you want to work.

In addition, Toad menu bars can configure themselves to how you work with Toad. As you work, Toad collects usage data on the commands you use most often. Menus personalize themselves to your work habits, moving the most used commands closer to the top of the list, and hiding commands that you use rarely.

Note:

Commands that have been removed from the toolbar and not the menu bar (or the other way around) do not display in the Unused area. Because of this, it may not be obvious that you have removed a command from one location and not the other.

5.1.5 To view new/removed commands

1. Right-click the toolbar/menu and select **Customize**.
2. Select the Commands tab.
3. To view new commands, select *[New]* in the **Categories** field.
4. To view commands you removed, select *[Unused]* in the **Categories** field.
5. To add a new/removed command to a menu/toolbar, drag the command to the toolbar/menu.

If you want to heavily modify an existing toolbar or menu, it may be easier to

create your own custom toolbar or menu instead.

5.1.6 To create a custom toolbar or menu

1. Right-click the toolbar/menu and select **Customize**.
2. Create the new toolbar/menu. Review the following for additional information:

To create a...	Complete the following:
toolbar	<ol style="list-style-type: none">a. Click New.b. Enter a name your new toolbar. A blank toolbar displays in the user interface below the existing toolbars.
menu	<ol style="list-style-type: none">a. Select the Commands tab.b. Select <i>New Menu</i> in the Categories field.c. Select <i>New Menu</i> in the Commands field and drag it to the menu bar where you want it located. The pointer changes to a vertical I-bar at the menu bar. <p>Tip: You can create sub-menus by dragging a new menu into an existing one.</p>

3. To add commands, select the **Commands** tab in the **Customize** window. Drag the command from the **Commands** field to the toolbar/menu. An I-bar pointer marks where the command will be dropped

Note: You can rearrange and rename the commands, toolbars, and menus.

4. To lock the toolbars, right-click a toolbar and select **Lock Toolbars**.

5.1.7 To customize the toolbar or menu

1. Right click the toolbar or menu and select **Customize**.
2. Change the toolbar or menu.

If you want to...	Complete the following:
Change the order of commands	Drag the item on the toolbar/menu to where you want it. An I- bar pointer marks where the command will be dropped.
Add commands	Complete the following: <ol style="list-style-type: none">a. Select the Commands tab in the Customize window.b. Drag the command from the Commands field to the toolbar/menu. An I-bar pointer marks where the command will be dropped.

Rename the toolbar, menu, or command	<p>Complete the following:</p> <ol style="list-style-type: none"> Right-click the icon or text on the item you want to change. Enter the new name in the Name field. If you want to define a hotkey, include an ampersand (&) before the letter you want to assign as the hotkey. <p>Note: These are not the same as Toad shortcut keys, but rather the underlined letter for keyboard navigation.</p>
Remove a command or menu	Right-click the item and select Delete .
<p>Tip: You can create sub-menus by dragging a new menu into an existing one.</p>	

- To have Toad menus configure themselves, select **Menus show recently used commands first** on the Options tab.

If you select this option, Toad collects usage data on the commands you use most often. Menus personalize themselves to your work habits, moving the most used commands closer to the top of the list, and hiding commands that you use rarely.

- To lock the toolbars, right-click a toolbar and select **Lock Toolbars**.

You can display additional menus, such as Team Coding or Create Objects.

5.1.8 To display additional menus

1. Right-click the menu bar and select **Customize**.
2. Select the Commands tab.
3. Select *Menus* in the **Categories** field.
4. Click the menu you want to add (for example, Team Coding) in the right pane and drag it to the menu bar where you want it located. The pointer changes to a vertical I-bar at the menu bar.

5.1.9 To change the toolbars you display

1. Right-click the toolbar area.
2. Select the toolbars you want to display, and clear the toolbars you want to hide.

5.1.10 To reset default toolbars and menus

» Right-click a toolbar and select **Restore defaults**.

It is possible to remove all the toolbars from the Editor. If this happens, you can restore the toolbars to your windows without resetting all the default settings.

5.1.11 To restore lost toolbars from the Editor

1. Right-click the Desktop panels tab area.
2. Select **Customize**.
3. Select the Toolbars tab.
4. Select the Editor toolbars you want to display.

Toad provides hundreds of options for you to customize its behavior. If there is a specific feature or behavior you would like to change, try

searching for it in the Options window.

5.1.12 Customize Shortcut Keys

Note: If you have customized your shortcut keys, you will not automatically be able to use new shortcuts added in future Toad upgrades. However, you can reset your shortcut keys to the default in order to gain access to all new shortcuts.

Menu hot keys are the keys that you access by pressing the ALT key and then the character in the menu item that is underlined to open that menu or command. You can configure the underlined character.

5.13 To change the hot key

1. Right-click the toolbar and select **Customize**.
2. Right-click the menu item you want to change.
3. Change the underlined character by changing the location of the ampersand in the **Name** field. For example, *&Tools* underlines the *T*, while *T&ools* underlines the *o*.

5.1.14 To change shortcut keys

1. Click  on the standard toolbar.

Tip: You can also select **View | Toad Options**.

2. Select **Toolbars/Menus | Shortcuts**.
3. Select the command for which you want to set or change the shortcut keys.
4. Type the keystrokes you want to use.

The shortcut key is changed as you type. If there is a conflict with another shortcut key, an asterisk (*) displays in the Conflict column. You can then find the conflict and remove it.

Note: This option only allows you to use one keystroke after a control key (such as CTRL or ALT).

Toad provides dozens of standard shortcut keys, plus you can assign new ones or customize the standard ones. Toad also allows you to print out your current list of shortcut keys.

General	Description
CTRL+D	Open Quick Describe window.
CTRL+TAB	Cycle through a collection of "child windows" or tabs in a window
F1	Open the Toad documentation
F4	Immediately describe object in popup window.
F10	Display right-click menu

Debugger	Description
CTRL+F5	Add watch at cursor
CTRL+ALT+B	Display the PL/SQL Debugger Breakpoints window
CTRL+ALT+D	Display the PL/SQL Debugger DBMS Output window
CTRL+ALT+E	Display the PL/SQL Debugger Evaluate/Modify window
CTRL+ALT+C	Display the PL/SQL Debugger Call Stack window

CTRL+ALT+W	Display the PL/SQL Debugger Watches window
F11	Run (continue execution)
F12	Run to cursor
SHIFT+F5	Set or delete a breakpoint on the current line
SHIFT+F7	Trace into
SHIFT+F8	Step over
SHIFT+F10	Trace out
SHIFT+CTRL+F9	Set parameters
Editor	Description
ALT+UP	Display previous statement
ALT+DOWN	Display next statement (after ALT+UP)
CTRL+B	Comment block
CTRL+E	Execute Explain Plan on the current statement
CTRL+M	Make code statement.
CTRL+N	Find sum of the selected fields. You can also include additional calculations, such as the average or count.
CTRL+P	Strip code statement.
CTRL+T	Display pick list drop-down
	There are a variety of shortcut keys to use with the pick list.
CTRL+F9	Verify statement without execution (parse) in the Editor
CTRL+F12	Pass the SQL or Editor contents to the specified external editor.
CTRL+PERIOD	Display code completion list

CTRL+ENTER	Execute current SQL (same as SHIFT+F9)
CTRL+ALT+PAGEUP	Navigate to the previous results panel tab
CTRL+ALT+PAGEDOWN	Navigate to the next results panel tab
F2	Toggle full screen Editor
F5	Execute as script.
F6	Toggle between Editor and Results panel
F7	Clear all text, trace into the Editor
F8	Recall previous SQL statement in the Editor
F9	Execute statement in the Editor
SHIFT+F2	Toggle full screen grid
Find and Replace	Description
CTRL+F	Find text
CTRL+G	Go to line number
CTRL+R	Find and replace
F3	Find next occurrence
SHIFT+F3	Find previous occurrence

5.1.15 Toad Insight Pick List Shortcuts

There are a variety of shortcuts you can use to display the pick list and make a selection. Toad also provides options for you to customize the pick list behavior

See "Code Assist Options" in the online help for more information.

General	Description
CTRL+T	Display pick list for object (name) at caret. If a stored alias exists by that name, then that alias' object is shown in the pick list.
CTRL+SHIFT+T	Display pick list for object (name) at caret. This option ignores aliases with the same name.
LEFT ARROW	Move the caret left while filtering the pick list.
RIGHT ARROW	Move the caret right while filtering the pick list.
Make Selection	Description
Double-click the selection	Insert the selection and close the pick list.
ENTER	Insert the selection and close the pick list.
PERIOD	Insert the selection and a period after it. The pick list remains open and displays child objects, if there are any.
SPACE	Insert the selection and a space after it.
TAB	<p>Insert the a partial selection if possible and leave the pick list open; if a partial selection is not possible, insert the selection and close the pick list.</p> <p>TAB accepts as much as possible without changing the list of displayed objects. For</p>

	example, if the pick list displays a list of columns that all start with MY_COL, Toad would insert MY_COL when you press TAB and leave the picklist open. If the columns did not have a common preface, Toad would insert the selected column and close the pick list.
(OPEN PARENTHESES	Insert the selection and "(" after it.
Close Pick List	Description
Click outside the pick	Close the pick list without making a selection.

list	
ESC	Close the pick list without making a selection.

You can print your list of shortcut keys to use as a reference.

[5.16 To print the list of shortcut keys](#)

1. Click  on the standard toolbar.

Tip: You can also select **View | Toad Options**.

2. Select **Toolbars/Menus | Shortcuts**.
3. Click the Category or Shortcut column to sort the list.

Click **Print**

5.2 Create and Manage Connections

To connect to a database server (referred to as "database"), Toad requires that you have a database client ("client") installed on your computer. A client is simply software that accesses the database through a network.

You can have multiple Oracle clients installed on your computer. These client locations are also referred to as Oracle homes, and you can select which one Toad currently uses on the Database Login window.

See the Release Notes for a complete list of the client and database versions that Toad supports.

Important: It is recommended that your client version be of the same release (or later) as your database server. This is an Oracle recommendation to prevent performance issues.

Connection Files

The Oracle client installation generally includes connection configuration files that are used to facilitate communication between your computer and the database. Toad uses the following connection configuration files, depending on the connection type you select:

Connection File	Description
------------------------	--------------------


SQLNET. ora	<p>Specifies configuration details for Oracle's networking software, such as trace levels, the default domain, session characteristics, and the connection methods that can be used to connect to a database (for example, LDAP and TNSNAMES). If a method is not listed, you cannot use it.</p> <p>Toad uses the SQLNET.ora file for all connection methods, and consequently you must be able to access this file for any connection method.</p>
TNSName s.ora	<p>Defines database address aliases to establish connections to them. Toad must be able to access the TNSNames.ora file for TNS connections.</p>

Conne ction File	Description
	<p>Note: If you have multiple Oracle clients installed or want to use a TNSNames.ora file on a network, you may want to use the TNS_NAMES environment variable to simplify managing TNS connections.</p>
LDAP.ora	<p>Defines directory access information using Lightweight Directory Access Protocol (LDAP). Toad must be able to access the</p>


5.2.1 Create New Connections

There are a few prerequisites you must have to connect to an Oracle database.

To create a new connection

1. Click  in the standard toolbar to open the Database Login window.

Note: You can also select **Session | New Connection**.

2. Click  on the Database Login toolbar. The Add Login Record window displays.

Note: Instead of creating the connection in the Add Login Record window, you can directly enter the connection information in the Database Login window. However, this method forces you to connect to the database, and you cannot enter some of the additional connection information until after you connect.

3. Complete the **User/Schema** and **Password** fields.
4. Select a connection method:

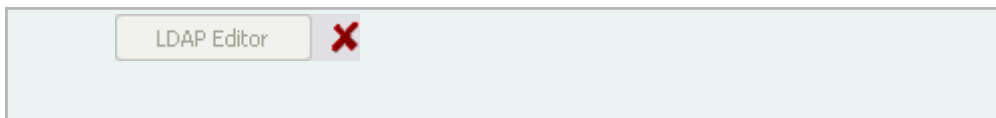
TNS	<p>Select a database in the Database field.</p> <p>Toad uses the listings in your TNSNames.ora file to populate the list.</p> <p>You can edit the TNSNames.ora file directly in Toad.</p> <p>Note: If you have multiple Oracle clients installed or want to use a TNSNames.ora</p>
-----	--

	<p>file on a network, you may want to use the TNS_ NAMES environment variable to simplify managing TNS connections.</p> <p>See "Create a Variable for the TNSNames.ora File" in the online help for more information.</p>
--	---


Direct	Enter the Host , Port , and either the Service Name or SID of the database to which you want to connect.
LDAP	Select the LDAP descriptor in the LDAP Descriptor field. You can edit the LDAP.ora file directly in Toad.

Notes:

- . Toad must be able to access the SQLNET.ora file to use any of the connection methods. Toad must also be able to access the LDAP.ora file for LDAP connections and the TNSNames.ora file for TNS connections.
- . If Toad cannot connect to one of these files, a red X displays beside the editor button for that file. For example, the following image indicates that Toad cannot access the LDAP.ora file. You would have to resolve the issue before you could make an LDAP connection.



5. Complete the remaining fields as necessary.

Connect as	Select the connection privilege level field.
Color	<p>Select a color to border windows that use the active connection.</p> <p>Note: The color displays in all Toad user interface elements that use the connection, which is very helpful when you have multiple active connections.</p>
Connect Using	<p>Select the Oracle home.</p> <p>Note: You can only connect to one Oracle home at a time. This field is disabled if you are already connected to a database.</p>
Alias	<p>Enter a description or Toad based alias or nickname for the connection.</p> <p>By default the alias only displays in the connections grid, but you can have Toad display the alias instead of the database name. To enable this option, select View Toad Options Windows and select the Use alias instead of database checkbox.</p>
Exec ute Acti	<p>Select to execute an action whenever Toad connects to the database. Then, click  by the Action field to select the action. See</p>

on	
----	--

upon Connect ion	<p>"Automation Designer Overview" in the online help for more information.</p> <p>You can also select a parameter file.</p> <p>Note: Toad only executes actions upon connection when you execute through the user interface. Toad does not execute actions when it is executed through command line.</p>
Custo m Colu mns	Complete the custom fields, if you have defined any.
Save Pass word	<p>Select to have Toad remember the password for only this connection.</p> <p>If Save passwords is selected in the Database Login window, then this field is selected by default.</p>
Auto Conn ect	Select to have Toad automatically make the selected connection on startup.
Favorite	Select this checkbox to mark the connection as one of your favorites. You can have the Database Login window only display your favorites by selecting Show favorites only

	at the bottom of the window.
Read Only	Select this checkbox to make the connection read only, meaning that you cannot make any changes to the database. This option is especially helpful when you want to access data for a production database but you do not want to accidentally make any changes.

1. Save the login record.

Review the following for additional information:

- . To save the record without connecting to the database, click **OK**
- . To save the record and connect to the database, select the **Connect** checkbox and click **OK**.
- . To save the record and reuse the field values to quickly enter new connections, click **Post**.

2. Optional: Manage multiple connections.

5.3 Manage Oracle Homes

Toad provides much of the same functionality as the Oracle Home Selector.

Tip: If comparing results in SQL*Plus and Toad for Oracle, confirm they are pointing to the same Oracle home.

Only one Oracle home can be in use at one time. This means that once a connection is made, all future connections will use the same Oracle home, regardless of default home listed for a connection. If you want to use a

different Oracle home for a new connection, you must close all open connections first.

Default homes can be assigned for a connection or for Toad. When a default Oracle home is assigned to a particular connection, any time you make that connection from the connection grid, Toad automatically uses that Oracle home. When a default Oracle home is assigned to Toad, Toad automatically uses that Oracle home any time you create a connection to a new database.


Toad searches for Oracle homes in several different ways.

5.3.1 Select an Oracle

Home Note:


- . If you have multiple Oracle clients installed or want to use a TNSNames.ora file on a network, you may want to use the TNS_NAMES environment variable to simplify managing TNS connections.

5.3.2 To select an Oracle home

1. Click  in the standard toolbar to open the Database Login window.

Note: You can also select **Session | New Connection**.


2. With no open connections, select an Oracle home in the **Connect using** field.

Note: To see more information about the home you have selected or change the SID, NLS_LANG, or SQLPATH, click  to open the Oracle Home Editor.

3. To set this as the default Oracle home for all connections, select **Make this the Toad default home**.

You must restart Toad to have changes made here take effect.


5.3.3 To edit the Oracle home

1. Click  beside the **Connect using** box on the Database Login window.
2. Select an Oracle home by clicking on its node. You can then:
 - . Click **Clipboard**. This will copy the selected information to the clipboard so you can paste it into an email, or another document.
 - . Click **Advice**. This will tell you if you have a proper Net8 installation for this home, or suggest changes to your installation.
 - . Right-click and choose to edit one of the following:
 - . SID for the selected home
 - . NLS_LANG for the selected home
 - . SQLPATH for the selected home

5.4 Edit Oracle Connection Files

From the SQLNET editor you can easily edit your SQLNET.ora parameters. The parameters on this window are standard Oracle parameters. See Oracle's documentation for more information.

5.4.1 To edit your SQLNET.ora file

1. Click  in the standard toolbar to open the Database Login window.

Note: You can also select **Session | New Connection**.

2. Click **SQLNET Editor**.

3. To back up your file before editing it, click **Create Backup File**.

Note: It is recommended that you create a backup file before you make any changes. This assures that if something goes wrong you can restore the original settings.

5.5 Execute and Manage Code

The Toad Editor lets you edit many types of statements and code, and Toad provides many options to customize the Editor's behavior.

5.5.1 Working in the Editor

The Editor attaches itself to the active connection in Toad, but if you do not have a connection you can still use it as a text editor. You can also change the schema to execute against from the Current Schema toolbar.

Tips:

The Editor's right-click menu contains many options to help you work with code. When you are trying to figure out how to do something, try right-clicking the Editor to see if it is available in the menu.

Select an object and press F4 to display the object's properties.

Select an object and press SHIFT+F4 to open the Action Console and select from the listed Actions related to that object.

If you press CTRL and click a PL/SQL object, the object opens in a new Editor tab. If you press CTRL and click a non-PL/SQL object, the object opens in the Describe Objects window.

The Editor is organized into the following areas:

Area	Description
Navigator Panel	The Navigator Panel is a desktop panel that displays an outline of the Editor contents in the active tab. You can click on the items listed to navigate to that statement in the Editor. The Navigator Panel is displayed on the left-hand side by default, but you can change where it is docked.
Editor	<p>The main Editor window displays code in separate tabs. You can create tabs for different bits of code, or different types of code. SQL and PL/SQL can go in the same tab. Toad can tell where the cursor is located and compile PL/SQL or run SQL as required.</p> <p>Note: If you have multiple statements in the Editor, you must trail them with a valid statement terminator such as a semi-colon.</p>
Desktop	The desktop panels contain many options for tab display, depending on what

Area	Description
Panels	kind of code you are working with and what you want to do with it. In addition, you can configure how these panels display to make Toad work for you.

5.5.2 Important Editor Settings

Toad provides many options to let you customize the Editor's behavior. Editor settings are controlled from:

View | Toad Options | Editor. The following table describes these options:

Option	Description	Navigate
Auto Replace	Lets you set substitution text phrases that correspond to replacement text. See Auto Replace Substitutions.	View Toad Options Editor Auto Replace
Code templates	Select code template settings. Code templates use a manual keystroke (CTRL+SPACE) to perform substitutions. See "Code Completion Templates" in the online help for more information.	View Toad Options Editor Code Templates

Commit after every statement	Commit every time a statement is run, after any posted edits are made in the grid, and after a row is deleted in the grid. Enabling this option makes it very easy to accidentally change or delete data. It is recommended that you do not select this option, and you should never have it enabled when you are working on a production database.	View Toad Options Oracle Transactions
Font and syntax highlighting	Select the Editor display font and select syntax highlighting settings.	View Toad Options Editor Font and Styling
Tab stops	Enter the number of spaces entered when you press TAB.	View Toad Options

Option	Description	Navigate
		Editor Behavior

When closing connections	Commit, rollback, or prompt when closing connections. This field is disabled if you select Commit after every statement . Selecting Commit makes it very easy to accidentally change or delete data. It is recommended that you select Prompt .	View Toad Options Oracle Transac tions
--------------------------	---	--

You can easily configure which panels display on your Editor desktop and where they display. You can select panels to display one at a time or in groups. When you have configured it, you can save the desktop with its own name, returning to it whenever the need arises. In addition, you can turn on Auto-save current desktop, and however you have the desktop set when you change tabs or close Toad will be how your desktop is defined the next time you open the Editor.

[5.5.3 To display panels one at a time](#)


1. Right-click the Editor and select **Desktop**.
2. Select the panel you want to display or hide.

[5.5.4 To configure your desktop](#)

1. Right-click the panel area near the bottom of the window.
2. Select **Desktop | Configure Desktop Layout**.
3. Select the panels you want to display in the Show column, and click the drop down menus in the Dock Site column to change where the panel is docked. By default, all except the Navigator will

be docked below the Editor.

5.5.5 To save your desktop

1. Click  on the Desktops toolbar.
2. Enter the name you want to use for this desktop.

5.5.6 To use a saved desktop

- » From the drop-down desktop menu, select the desktop you want to use.

5.5.7 To restore a desktop

- » Click the drop-down arrow on  and select **Revert to Last Saved Desktop** or **Restore Default Desktop**.

You can split the Editor to easily compare code revisions.

Tip: To remove the split layout, right-click in the Editor and select **Split Editor Layout**

| **Not Split**.

5.5.8 To split the Editor

1. Right-click the Editor and select **Split Editor Layout**.
2. Select **Left-Right** or **Top-Bottom**.

5.6 Execute Statements and Scripts

Toad provides many different options for you to execute scripts:

5.6.1 Execute Single Statements

You can easily execute a single statement in the Editor. Toad's parser

identifies and executes the statement or compiles the PL/SQL at the cursor.

Note: If you select code and execute, Toad ignores the parser results and executes the portion that is selected. This may cause errors, especially if you select more than one statement. It is better to place your cursor in the statement you want to execute and let Toad select the statement.

This method fetches matching records in batches to improve performance.


Notes:

Executing a statement can produce editable data.

Toad provides several options to execute a full script or multiple statements.

You can easily execute a SQL statement embedded within PL/SQL.

5.6.2 To execute a statement in the Editor

» Place the cursor in the statement and click  on the Execute toolbar (F9).

Note: To cancel the execution, click  in the Execute toolbar.

The Current Schema drop-down lets you work with a schema other than the one to which you are connected. This can be useful if, for example, you have tested a SQL statement in your test schema and now want to execute it on several other schemas without disconnecting and reconnecting.

By default, the current schema is set to your current connection. When you

use this drop-down, Toad issues an `ALTER SESSION SET current_schema` command. After you execute, Toad issues the `ALTER SESSION SET current_schema` command again to return to the original connection schema.

Notes:

- You must have the `ALTER SESSION` system privilege to use this feature. If you do not have the privilege, the drop-down is disabled.
- Using this feature eliminates the need to prefix every table name with a schema name, and helps to eliminate ORA-00942 “table not found” errors.

5.6.3 To change the current schema

» Select a different schema in the Current Schema toolbar. The Current Schema drop-down does not work with script execution or debugging commands. However, because Execute as Script is designed to mimic SQL*Plus, you can use a set schema command to change the schema.

5.6.4 To change the schema in scripts

» Include the following command at the beginning of your script:

```
ALTER SESSION SET current_schema = "USERNAME"
```

You can save SQL statements and easily insert them into the Editor at any time. The best way to save SQL statements is with the Named SQL feature. Toad also allows you to export and import your saved SQL.

Toad lists saved and recently executed SQL statements in the SQL Recall pane.

Notes:

- . If you want a quicker way to save SQL statements, you can save them as Personal SQL statements by selecting **Editor | Add to Personal SQLs**. This bypasses the dialog to name the SQL. However, the only way to reuse Personal statements is from the SQL Recall pane.
- . Toad stores all saved SQL in `User Files\SavedSQL.dat`.

5.6.5 To save statements from the Editor

1. Select the statement in the Editor.
2. Select **Editor | Add to Named SQLs**.
3. Enter a name for the SQL statement.

Note: The name is case sensitive. For example, you can save both "sql1" and "SQL1".

5.6.6 To use a saved statement in the Editor

1. Select one of the following options:
 - . Press **CTRL+N** in the Editor and select the statement from the pick list.
 - . Enter `^MyNamedSQL` in the Editor, where *MyNamedSQL* is the name of your saved SQL statement. Toad replaces the SQL name with the saved statement at execution.
 - . Double-click or drag the statement from the SQL Recall pane.

5.6.7 To view saved statements

- » Select **View | SQL Command Recall | Named**.

5.6.8 To edit statements in the SQL Recall pane

- » Select a statement and click  on the SQL Recall toolbar.

Toad saves recently executed statements in the History tab of the SQL Recall pane. This list is organized with the most recent SQL at the top by default. You can select a statement from this list and run it, save the statement for easy recall, or remove a statement from this list.

The SQL Recall pane also lists your saved SQL statements in the Named and Personal tabs.

Note: You can change the number of statements that SQL Recall saves in the History (500 is default) or save only SQL statements that executed successfully. You can select these options and other SQL Recall settings on the Code Assist options page.

5.6.9 To view previously executed SQL statements

- » Select **View | SQL Command Recall | History (F8)**.

Tip: You can also press ALT+UP ARROW or ALT+DOWN ARROW in the Editor.

5.6.10 To open SQL statement directly in the Editor

- » Double-click or drag the statement from the SQL Recall pane.

5.6.11 To save statements in the History tab

1. Select a statement and click  in the SQL Recall toolbar.

2. Select *Named* in the **Type** field and enter a name for the statement in the **Name** field.

5.6.12 To edit statements in the SQL Recall pane

- » Select a statement and click  on the SQL Recall toolbar.


5.7 Format Code

You can have Toad format your code in the Editor. You can customize how Toad formats the code, such as inserting spaces instead of tabs or changing the case for SQL commands.

See "Formatter Options" in the online help for more information.

Note: You can format multiple scripts at one time from the Project Manager. See "Format Files" in the online help for more information.

5.7.1 To format a statement

- » Select the statement you want to format and click 'Format Code'  on the Editor toolbar, or select the drop-down arrow to select:
 - Format Case Only,
 - Profile Code, *or*
 - Formatter Options (opens the Options window).

5.7.2 To format an entire script

- » Click  on the Edit toolbar.

5.8 Display Pick List (Automatically Complete Code)

The Toad Insight feature helps you write code by displaying a pick list with relevant object or column names. For example, if you start typing `SYS` and invoke the pick list, the `SYSTEM` user would be included in the pick list.

Toad provides options for you to customize Code Insight's behavior, such as adjusting the length of time before the pick list displays.

See "Code Assist Options" in the online help for more information.

5.8.1 To display the pick list

» Press CTRL+T, or begin typing a name and pause 1.5 seconds.

Note: There are additional shortcut keys you can use with Toad Insight.

You can extract a procedure from existing code into a new stored procedure or locally defined procedure.

Creating the new procedure and call depend heavily on the parser to determine which identifiers in the text selection must be declared as parameters in the new procedure. If Toad cannot parse the code, no extraction occurs.

5.8.2 To extract procedures

1. Select the code you want to extract in the Editor.
2. Right-click and select **Refactor | Extract Procedure**.
3. Select a procedure type.

Note: If you select stored procedure, you can choose to either include the "CREATE OR REPLACE" in the DDL instead of just "CREATE".

4. Enter the procedure name.

Tip: The new procedure and the resulting procedure call are created and inserted so that the code is syntactically correct, but no formatting is done to the code. You can have Toad format the code by pressing SHIFT+CTRL+F.

These commands add or remove comments from the selected block of text by adding or removing "--" from the beginning of each line.

5.8.3 To comment code

1. Select the code block.
2. Right-click and select **Refactor | Comment Block**.

Tip: You can also press CTRL+B.

5.8.4 To uncomment code

1. Select the code block.
2. Right-click and select **Refactor | Uncomment Block**.

Tip: You can also press SHIFT+CTRL+B.

Toad can find unused variables and identifiers in PL/SQL with code refactoring. If Toad finds unused variables, it displays the variables and lets you jump to the occurrence in the Editor.

Notes:

- . Toad only searches the object in the Editor, and does not evaluate

other PL/SQL objects that may reference it. Be careful when removing unused variables from package specifications, because they may be referenced in other PL/SQL that is not searched.

5.8.5 To find unused variables

1. Right-click code in the Editor.
2. Select **Refactor | Find Unused Variables**.

You can easily rename identifiers (variables, parameters, or PL/SQL calls) for PL/SQL in the Editor with code refactoring.

Notes: Toad only searches the PL/SQL object in the Editor. Be careful when renaming identifiers in package specifications, as they maybe be referenced in other PL/SQL that is not searched.

5.8.6 To rename identifiers

1. Right-click an identifier in the Editor and select **Refactor | Rename Identifier**.
2. Enter the new name in the **Name** field.

5.9 Debug PL/SQL

You can debug PL/SQL, SQL scripts, and Java in Toad. Toad's documentation includes tutorials on how to debug. See "Debugging a Procedure or Function Tutorial" in the online help for more information.

Notes:

- . There are minimum Oracle database requirements for using this feature.

- . The debugger is not designed to work with word-wrapped lines, since the Editor will then have a different set of line numbers than what is stored in Oracle. Toad provides a warning message about this if you open the procedure Editor while word-wrapping is enabled. To disable word-wrap, select **View | Toad Options | Editor | Behavior** and clear **Word wrap**.

5.9.1 Types of Debugging

Debugging in Toad requires you to select one type of debugging at a time for all database instances open per instance of Toad. For example, if you have three database connections in one instance of Toad, they must all be in the same debugging state. If you then opened another instance of Toad, with the same or different connections, they could be in a different debugging state.

DBMS Debug ger	<p>Debugs PL/SQL. Using the Debugger, you can set breakpoints, watches, and see call stacks. In addition, you can view DBMS output.</p> <p>Note: When using the PL/SQL Debugger and connecting to a RAC instance, you must have the TNSNAMES entry for the instance with the server directed the use connection or session here. Or, you must connect directly to an instance of the cluster without letting the server assign an instance.</p>
Script	Debugs SQL scripts. You can set breakpoints, run to cursor, step over, trace

Debugger	into, and halt execution of your scripts.
Attach External Session	External debugging allows you to debug PL/SQL that is run from an external session, such as another Toad window, SQL*Plus, or any other development tool which calls Oracle stored procedures.

You can also use Toad's Auto Debugger, which automatically inserts DBMS_OUTPUT.PUT_LINE statements into the DDL. Once you compile the code and inspect the contents of the DBMS_OUTPUT buffer, you can remove all instances of DBMS_OUTPUT.PUT_LINE with the click of a button. See "Automatically Insert DBMS_OUTPUT Statements (Auto Debugger)" in the online help for more information.


5.9.2 Compile with Debug Information

To use the debugger fully with PL/SQL or Java packages, you need to compile your object with debug information. If you have not compiled with debug information, in databases in versions before 10g you can step into a unit, step over and so on, but you cannot see watches unless the object is compiled with debug. In a 10g database you cannot step into code or step over unless the object was compiled with debug. You can only execute.

In addition, if you are debugging an object that has dependent objects, you cannot step into the dependents unless they, too, are compiled with debug information.

See "Dependencies and References" in the online help for more information.

5.9.3 To enable compile with debug



» Click  on the main toolbar or select **Session | Toggle Compiling with Debug**.

Note: You can have Toad enable **Toggle Compiling with Debug** by default for each new session.

See "Execute and Compile Options" in the online help for more information.

You can debug PL/SQL objects in the Editor. When you open a complete package or type in the Editor, the spec and body open in separate tabs by default. However, Toad provides options to control how objects are split, reassembled, and saved.

5.9.4 To start the Debugger

1. Open a PL/SQL object in the Editor.
2. Click on the  main toolbar or select **Session | Toggle Compiling with Debug**. This enables debugging.
3. Compile the object on the database.
4. Select one of the following options on the Execute toolbar to begin debugging:
 - . Execute PL/SQL with debugger ()
 - . Step over
 - . Step into
 - . Run to cursor



5.9.5 View DBMS Output

Oracle provides a specifically designed package called DBMS_OUTPUT with functions for debugging PL/SQL code. It uses a buffer that your PL/SQL code writes into and then a separate process queries the buffer out and displays the contents.

You must enable DBMS Output before executing the PL/SQL. In Toad, output displays after the procedure has completed execution, not while you are stepping through the code. In nested procedure calls, all procedures must have run to completion before any DBMS Output content is displayed.

Troubleshooting

If you do not see DBMS Output, try the following suggestions:

- . Right-click the lower pane and select **Desktop Panels | DBMS Output**.
- . Make sure the **Toggle Output On/Off** button is on () in the DBMS Output tab. Then, set the interval in the Polling Frequency box. If the toggle is on, Toad periodically scans for and displays DBMS Output content.
- . Contact your Oracle DBA to make sure the DBMS_OUTPUT package is enabled on your database.

5.9.6 Debug External Sessions

This feature is extremely useful when the external session calls a stored procedure with complex parameters, such as cursors, that are not easily

simulated from Toad. Rather than trying to simulate the complex environment within Toad, you can simply connect to the external application and then debug the code in its native environment.

5.9.7 To debug an external session

1. Prepare the external session:

- a. Disable server output on the external session by executing the following command:

```
set serveroutput off
```

Note: If server output capture is enabled, Oracle freezes on calls to the DBMS_OUTPUT package.

- b. Execute the following command:

```
id =  
  
dbms_debug.initialize('TOAD  
  
' ) dbms_debug.debug_on;
```

where TOAD can be replaced by any ID string.

- c. Execute the PL/SQL.

2. Attach to the external session and begin debugging in Toad:

- a. In the Editor, connect to the same database instance as the external application.
- b. Compile the PL/SQL with debug enabled. See "You can debug PL/SQL objects in the Editor. When you open a complete package or type in the Editor, the spec and body open in separate tabs by default. However, Toad provides options to control how objects

are split, reassembled, and saved. " (page 51) for more information.

- c. Select **Debug | Attach External Session**.
- d. Enter the ID specified in the initialize statement (in step 1b) and click **OK**. Toad begins debugging the procedure called from the external session.

5.9.8 Troubleshoot External Debugging

There may be cases where the debug ID is set incorrectly, or some other error may hang the debugger. In this case it is necessary to clear out any open debug pipes from the database.

Pipes may be viewed in the V\$DB_PIPES dynamic view, and removed with the following procedure when executed as SYS:

```
CREATE OR REPLACE PROCEDURE drop_pipe
```

```
(p_pipename IN VARCHAR2) AS
```

```
x NUMBER;
```

```
BEGIN
```

```
x := SYS.DBMS_PIPE.remove_pipe
```

```
(p_pipename); DBMS_OUTPUT.put_line (
```

```
'Drop Pipe ' || p_pipename || DECODE (x, 1, ' 
```

```
SUCCESS', ' FAILED'))); END;
```

```
/
```

Disable Debugging in External Session

Ensure that debugging in the external session has been disabled. After the external application finishes execution, it should execute the command:

```
dbms_debug.debug_off
```

Otherwise, all subsequent PL/SQL that this application submits for execution will be run in debug mode. This causes the application to hang until Toad attaches to it again.

5.10 Toad|PLSQL

PL/SQL is a combination of SQL along with the procedural features of programming languages. It was developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL. PL/SQL is one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java. This tutorial will give you great understanding on PL/SQL to proceed with Oracle database and other advanced RDBMS concepts.

Audience

This tutorial is designed for Software Professionals, who are willing to learn PL/SQL Programming Language in simple and easy steps. This tutorial will give you great understanding on PL/SQL Programming concepts, and after completing this tutorial, you will be at an intermediate level of expertise from where you can take yourself to a higher level of expertise.

Prerequisites

Before proceeding, you should have a basic understanding of software basic concepts like what is database, source code, text editor and execution of programs, etc. If you already have an understanding on SQL and other computer programming language, then it will be an added advantage to proceed.

Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.INSERT_GROUPS_CLASSES)]

File Edit Search Editor Session Database Debug View Utilities Window Help

Search Toad World

Editor Editor Schema Browser

ZAG_SCHED

Functions Procedures Packages

INSERT_GROUPS_CLASSES

INSERT_GROUPS_PERIODS

INSERT_LECTURES_PERIODS

INSERT_LECTURES

INSERT_SECTIONS

INSERT_SECTIONS_CLASSES

INSERT_SECTIONS_PERIODS

INSERT_TIMING_CURSOR

REMOVE_LECTURES_SECTIONS

```
1 CREATE OR REPLACE PROCEDURE ZAG_SCHED.INSERT_GROUPS_CLASSES
2 IS
3     CURSOR GROUPS_CURSOR IS
4         SELECT * FROM GROUPS
5         ORDER BY GROUP_ID ASC;
6
7     CURSOR CLASSES_CURSOR IS
8         SELECT * FROM CLASSES
9         WHERE CLASS_TYPE_ID = 1
10        ORDER BY CLASS_ID ASC;
11 BEGIN
12     FOR GROUPS_REC IN GROUPS_CURSOR LOOP
13         FOR CLASSES_REC IN CLASSES_CURSOR LOOP
14             INSERT INTO GROUPS_CLASSES
15                 (GROUP_ID, CLASS_ID)
16             VALUES
17                 (GROUPS_REC.GROUP_ID, CLASSES_REC.CLASS_ID);
18         END LOOP;
19     END LOOP;
20 END;
```

Line: 9 ZAG_SCHED@XE

AutoCommit is OFF [CAPS NUM] [TNS]

Links This PC

ENG 9:28 PM

Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.INSERT_GROUPS_PERIODS)]

File Edit Search Editor Session Database Debug View Utilities Window Help

Search Toad World

Editor Editor Schema Browser

ZAG_SCHED

Functions Procedures Packages

INSERT_GROUPS_CLASSES

INSERT_GROUPS_PERIODS

INSERT_LECTURES_PERIODS

INSERT_LECTURES

INSERT_SECTIONS

INSERT_SECTIONS_CLASSES

INSERT_SECTIONS_PERIODS

INSERT_TIMING_CURSOR

REMOVE_LECTURES_SECTIONS

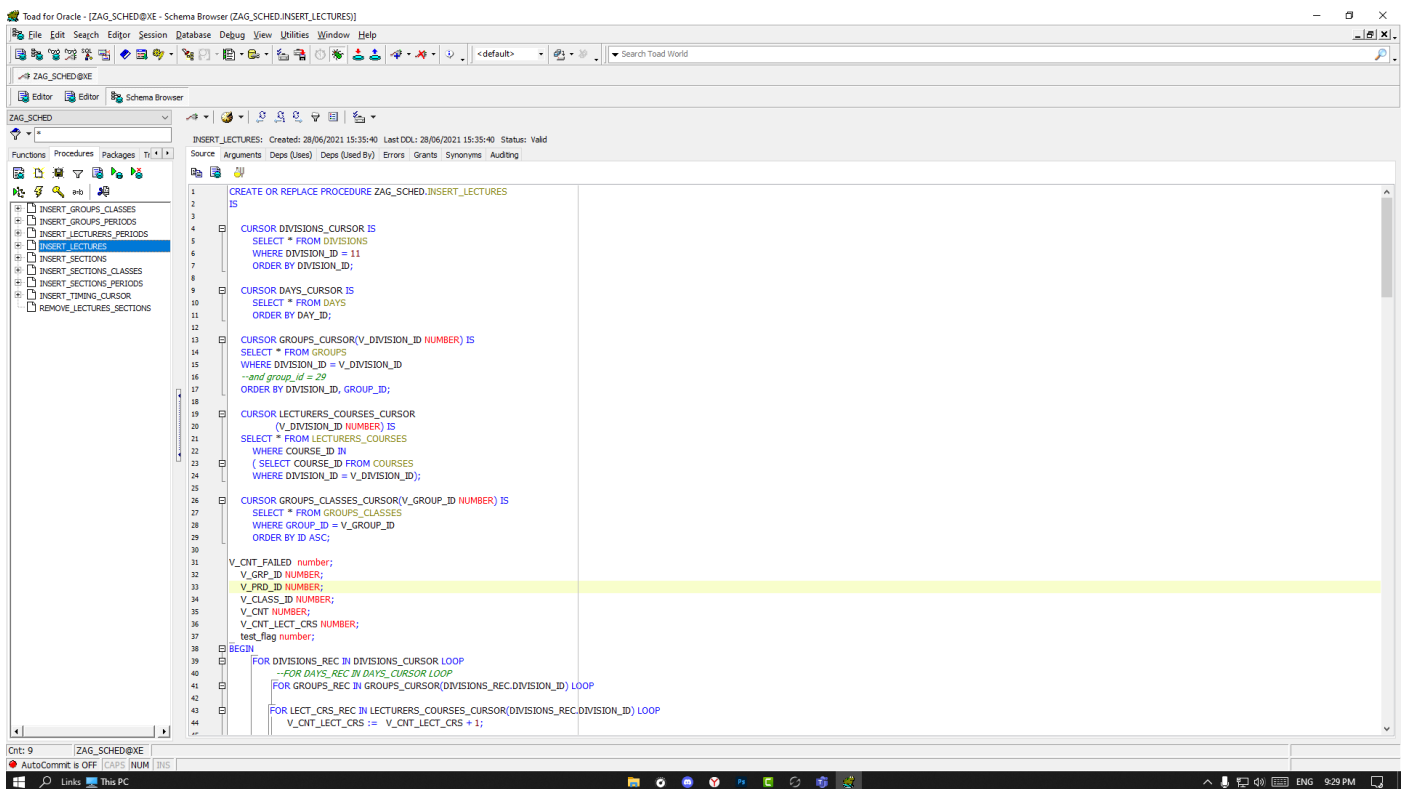
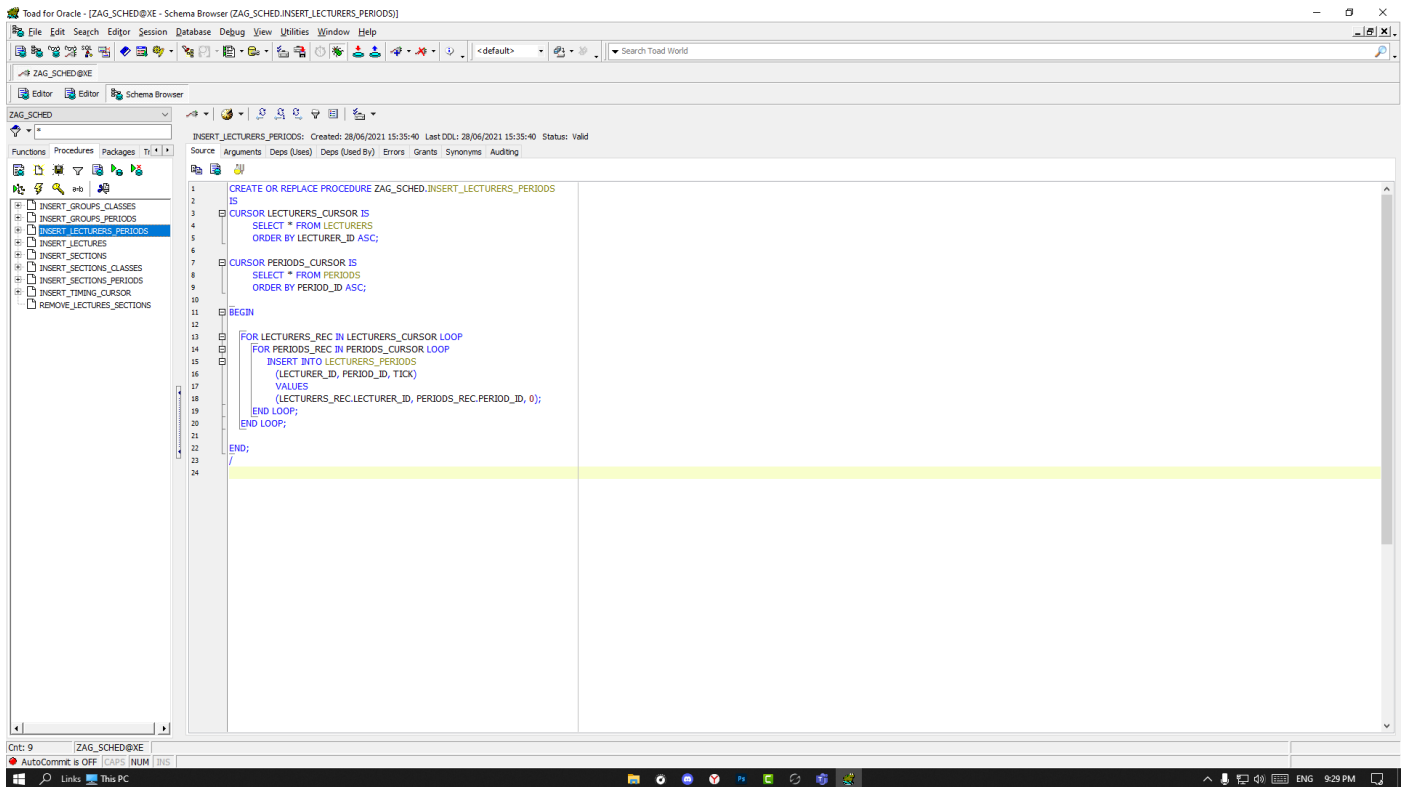
```
1 CREATE OR REPLACE PROCEDURE ZAG_SCHED.INSERT_GROUPS_PERIODS
2 IS
3     CURSOR GROUPS_CURSOR IS
4         SELECT * FROM GROUPS
5         ORDER BY GROUP_ID ASC;
6
7     CURSOR PERIODS_CURSOR IS
8         SELECT * FROM PERIODS
9         ORDER BY PERIOD_ID ASC;
10 BEGIN
11     FOR GROUPS_REC IN GROUPS_CURSOR LOOP
12         FOR PERIODS_REC IN PERIODS_CURSOR LOOP
13             INSERT INTO GROUPS_PERIODS
14                 (GROUP_ID, PERIOD_ID, TICK)
15             VALUES
16                 (GROUPS_REC.GROUP_ID, PERIODS_REC.PERIOD_ID, 0);
17         END LOOP;
18     END LOOP;
19 END;
```

Line: 9 ZAG_SCHED@XE

AutoCommit is OFF [CAPS NUM] [TNS]

Links This PC

ENG 9:29 PM



Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.INSERT_LECTURES)]

File Edit Search Editor Session Database Debug View Utilities Window Help

Search Toad World

Editor Schema Browser

ZAG_SCHED

Functions Procedures Packages

INSERT_GROUPS_CLASSES
INSERT_GROUPS_PERIODS
INSERT_LECTURES
INSERT_SECTIONS
INSERT_SECTIONS_CLASSES
INSERT_SECTIONS_PERIODS
INSERT_TIMING_CURSOR
REMOVE_LECTURES_SECTIONS

```

61  -- 2) VP SELECT A FIRST SINGLE CLASS SUITABLE FOR THIS GRP GROUPS_CLASSES
62  <<GROUPS_PERIODS_loop>>
63  LOOP
64  SELECT GROUP_ID, PERIOD_ID -- MUST BE 1 RECORD FOR EACH GROUP WITH SPECIFIC PRD : 1 LEC PER DAY
65  INTO V_GRP_ID, V_PRD_ID
66  FROM GROUPS_PERIODS
67  WHERE GROUP_ID = GROUPS_REC.GROUP_ID
68  AND PERIOD_ID IN ( SELECT PERIOD_ID FROM LECTURERS_PERIODS
69  WHERE LECTURER_ID = LECT_CRS_REC.LECTURER_ID
70  AND TICK = 0)
71  AND TICK = 0
72  AND ROWNUM = 1
73  ORDER BY ID ASC;
74  test_flag := 0;
75  FOR GROUPS_CLASSES_REC
76  IN GROUPS_CLASSES_CURSOR( GROUPS_REC.GROUP_ID) LOOP
77
78  test_flag := test_flag + 1;
79  BEGIN
80  -- REMAINDER : TO CHECK CLASS IS NOT
81  -- TAKEN BEFORE WITH THIS TIMER PERIOD
82  dbms_output.put_line('test flag = '||test_flag);
83  SELECT CLASS_ID
84  INTO V_CLASS_ID
85  FROM GROUPS_CLASSES
86  WHERE GROUP_ID = GROUPS_REC.GROUP_ID
87  AND CLASS_ID NOT IN
88  ( SELECT CLASS_ID FROM CLASSES_PERIODS -- LECTURES CLASSES WITH PERIODS
89  WHERE PERIOD_ID = V_PRD_ID)
90  AND ROWNUM = 1
91  ORDER BY ID ASC;
92  -- IF FOUND THEN
93  dbms_output.put_line('to be inserted - exit now');
94  EXIT GROUPS_PERIODS_loop;
95
96  EXCEPTION
97  WHEN NO_DATA_FOUND THEN
98  dbms_output.put_line('n d f '||LECT_CRS_REC.ID||', class = '||V_CLASS_ID||
99  ' prd = '||v_prd_id||', lecturer id = '||LECT_CRS_REC.LECTURER_ID||);
100  -- CONTINUE;
101  WHEN OTHERS THEN
102  dbms_output.put_line('OTHER');
103  END;
104  END LOOP;

```

Cre: 9 ZAG_SCHED@XE

AutoCommit is OFF [CAPS NUM] [TMS]

Links This PC

ENG 9:29 PM

Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.INSERT_LECTURES)]

File Edit Search Editor Session Database Debug View Utilities Window Help

Search Toad World

Editor Schema Browser

ZAG_SCHED

Functions Procedures Packages

INSERT_GROUPS_CLASSES
INSERT_GROUPS_PERIODS
INSERT_LECTURES
INSERT_SECTIONS
INSERT_SECTIONS_CLASSES
INSERT_SECTIONS_PERIODS
INSERT_TIMING_CURSOR
REMOVE_LECTURES_SECTIONS

```

106  IF test_flag = 4 then -- loop GROUPS_CLASSES finished with no class
107  -- available for this period
108  UPDATE GROUPS_PERIODS
109  SET TICK = 1
110  WHERE GROUP_ID = GROUPS_REC.GROUP_ID
111  AND PERIOD_ID IN ( SELECT PERIOD_ID FROM LECTURERS_PERIODS
112  WHERE LECTURER_ID = LECT_CRS_REC.LECTURER_ID
113  AND TICK = 0)
114  AND TICK = 0
115  AND ROWNUM = 1;
116  continue;
117  end if;
118
119  END LOOP GROUPS_PERIODS_loop;
120
121
122  -- Check the same grp with same crs
123  -- not to be inserted again in table lectures
124  SELECT COUNT(*)
125  INTO V_CNT FROM LECTURES
126  WHERE GROUP_ID = GROUPS_REC.GROUP_ID
127  AND COURSE_ID = LECT_CRS_REC.COURSE_ID
128  AND LECTURE_TYPE_ID = 1;
129
130  IF V_CNT = 1 THEN
131  V_CNT := 0;
132  dbms_output.put_line('V CNT = 0 '||LECT_CRS_REC.ID);
133  CONTINUE;
134  END IF;
135
136  -- INSERT INTO LECTURES ( 1 row per loop )
137
138  INSERT INTO LECTURES
139  (LECTURE_TYPE_ID, GROUP_ID,
140  COURSE_ID, LECTURER_ID)
141  VALUES
142  (1, GROUPS_REC.GROUP_ID, LECT_CRS_REC.COURSE_ID,
143  LECT_CRS_REC.LECTURER_ID);
144
145
146
147
148  SELECT COUNT(*)
149  INTO V_CNT_FAILED
150  FROM LECTURES_FAILED;

```

Cre: 9 ZAG_SCHED@XE

AutoCommit is OFF [CAPS NUM] [TMS]

Links This PC

ENG 9:29 PM

Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.INSERT_LECTURES)]

File Edit Search Editor Session Database Debug View Utilities Window Help

Search Toad World

ZAG_SCHED@XE

Editor Editor Schema Browser

ZAG_SCHED

Functions Procedures Packages

INSERT_GROUPS_CLASSES
INSERT_GROUPS_PERIODS
INSERT_LECTURES_PERIODS
INSERT_LECTURES
INSERT_SECTIONS
INSERT_SECTIONS_CLASSES
INSERT_SECTIONS_PERIODS
INSERT_TIMING_CURSOR
REMOVE_LECTURES_SECTIONS

```
148 SELECT COUNT(*)  
149 INTO V_CNT_FAILED  
150 FROM CLASSES_PERIODS  
151 WHERE PERIOD_ID = V_PRD_ID  
152 AND CLASS_ID = V_CLASS_ID;  
153  
154  
155 INSERT INTO CLASSES_PERIODS -- ( 1 row per loop )  
156 ( PERIOD_ID, CLASS_ID, LECTURE_ID )  
157 VALUES  
158 ( V_PRD_ID, V_CLASS_ID, LECTURES_SEQ.CURRVAL );  
159  
160  
161 IF V_CNT_FAILED > 0 THEN  
162 DBMS_OUTPUT.PUT_LINE('Failed = ' || CLASSES_PERIODS_SEQ.CURRVAL);  
163 end if;  
164  
165 -- THIS PERIOD WITH THIS GROUP IS TAKEN NOW  
166  
167 UPDATE GROUPS_PERIODS  
168 SET TICK = 1  
169 WHERE GROUP_ID = V_GRP_ID  
170 AND PERIOD_ID = V_PRD_ID;  
171  
172 -- THIS PERIOD WITH THIS LECTURER IS TAKEN  
173 UPDATE LECTURERS_PERIODS  
174 SET TICK = 1  
175 WHERE LECTURER_ID = LECT_CRS_REC.LECTURER_ID  
176 AND PERIOD_ID = V_PRD_ID;  
177  
178 END LOOP; -- END LECTURES_COURSES  
179  
180 DBMS_OUTPUT.PUT_LINE(V_CNT_LECT_CRS ' || V_CNT_LECT_CRS);  
181 V_CNT_LECT_CRS := 0;  
182 DBMS_OUTPUT.PUT_LINE(' ');  
183 END LOOP; -- END GROUPS  
184 -- END LOOP; -- END DAYS  
185 END LOOP; -- END DIVISIONS_CURSOR  
186  
187 commit;  
188  
189
```

Line: 9 ZAG_SCHED@XE

AutoCommit is OFF [CAPS NUM] [TMS]

Links This PC

ENG 9:30 PM

Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.INSERT_SECTIONS)]

File Edit Search Editor Session Database Debug View Utilities Window Help

Search Toad World

ZAG_SCHED@XE

Editor Editor Schema Browser

ZAG_SCHED

Functions Procedures Packages

INSERT_GROUPS_CLASSES
INSERT_GROUPS_PERIODS
INSERT_LECTURES_PERIODS
INSERT_LECTURES
INSERT_SECTIONS
INSERT_SECTIONS_CLASSES
INSERT_SECTIONS_PERIODS
INSERT_TIMING_CURSOR
REMOVE_LECTURES_SECTIONS

```
1 CREATE OR REPLACE PROCEDURE ZAG_SCHED.INSERT_SECTIONS  
2 IS  
3  
4 CURSOR DIVISIONS_CURSOR IS  
5 SELECT * FROM DIVISIONS  
6 WHERE DIVISION_ID = 13  
7 ORDER BY DIVISION_ID;  
8  
9 CURSOR DAYS_CURSOR IS  
10 SELECT * FROM DAYS  
11 ORDER BY DAY_ID;  
12  
13 CURSOR SECTIONS_CURSOR(V_DIVISION_ID NUMBER) IS  
14 SELECT * FROM SECTIONS  
15 WHERE GROUP_ID IN ( SELECT GROUP_ID FROM GROUPS  
16 WHERE DIVISION_ID = V_DIVISION_ID )  
17 ORDER BY SECTION_ID, GROUP_ID;  
18  
19 CURSOR LECTURERS_COURSES_CURSOR  
20 (V_DIVISION_ID NUMBER) IS  
21 SELECT * FROM LECTURERS_COURSES  
22 WHERE COURSE_ID IN  
23 ( SELECT COURSE_ID FROM COURSES  
24 WHERE DIVISION_ID = V_DIVISION_ID )  
25 AND LECTURER_ID IN ( SELECT LECTURER_ID FROM LECTURERS  
26 WHERE LECTURER_POSITION_ID = 2 );  
27 -- AND LECTURER_ID = 81;  
28  
29 CURSOR SECTIONS_CLASSES_CURSOR(V_SECTION_ID NUMBER) IS  
30 SELECT * FROM SECTIONS_CLASSES  
31 WHERE SECTION_ID = V_SECTION_ID  
32 ORDER BY ID ASC;  
33  
34 CURSOR LECTURER_SECTIONS_CURSOR(V_LECTURER_ID NUMBER) IS  
35 SELECT * FROM LECTURERS_SECTIONS  
36 WHERE LECTURER_ID = V_LECTURER_ID  
37 -- WHERE LECTURER_ID = 81  
38 -- WHERE SECTION_ID IN ( 1, 43, 48, 49 )  
39 ORDER BY SECTION_ID;  
40  
41 V_CNT_FAILED number;  
42 V_SECTION_ID number;  
43 V_PRD_ID number;  
44 V_CLASS_ID number;  
45
```

Line: 9 ZAG_SCHED@XE

AutoCommit is OFF [CAPS NUM] [TMS]

Links This PC

ENG 9:30 PM

Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.INSERT_SECTIONS)]

File Edit Search Editor Session Database Debug View Utilities Window Help

ZAG_SCHED@XE

Schema Browser

INSERT_SECTIONS: Created: 28/06/2021 15:35:40 Last DDL: 28/06/2021 15:35:40 Status: Valid

Source Arguments Deps (Used) Deps (Used By) Errors Grants Synonyms Auditing

```
49 BEGIN
50 FOR DIVISIONS_REC IN DIVISIONS_CURSOR LOOP
51     --FOR DAYS_REC IN DAYS_CURSOR LOOP
52     --FOR SECTIONS_REC IN SECTIONS_CURSOR(DIVISIONS_REC.DIVISION_ID) LOOP
53
54     FOR LECT_CRS_REC IN LECTURERS_COURSES_CURSOR(DIVISIONS_REC.DIVISION_ID) LOOP
55
56         V_CNT_LCT_CRS := V_CNT_LCT_CRS + 1;
57
58         FOR LECTURER_SECTIONS_REC IN
59             LECTURER_SECTIONS_CURSOR(LECT_CRS_REC.LECTURER_ID) LOOP
60
61             V_CNT_TEST := V_CNT_TEST + 1;
62             DBMS_OUTPUT.PUT_LINE(V_CNT_TEST || V_CNT_TEST);
63
64             -- <<SECTIONS_PERIODS_loop>>
65
66             -- LOOP
67             SELECT SECTION_ID, PERIOD_ID -- MUST BE 1 RECORD FOR EACH GROUP WITH SPECIFIC PRD : 1 LEC PER DAY
68             INTO V_SECTION_ID, V_PRD_ID
69             FROM SECTIONS_PERIODS
70             WHERE SECTION_ID = LECTURER_SECTIONS_REC.SECTION_ID
71             AND PERIOD_ID IN (SELECT PERIOD_ID FROM LECTURERS_PERIODS
72                             WHERE LECTURER_ID = LECT_CRS_REC.LECTURER_ID
73                             AND TICK = 0)
74             AND TICK = 0
75             AND ROWNUM = 1
76             ORDER BY ID ASC;
77
78             test_flag := 0;
79
80             FOR SECTIONS_CLASSES_REC
81                 IN SECTIONS_CLASSES_CURSOR(LECTURER_SECTIONS_REC.SECTION_ID) LOOP
82
83                 test_flag := test_flag + 1;
84
85                 --begin
86                 -- REMAINDER : TO CHECK CLASS IS NOT
87                 -- TAKEN BEFORE WITH THIS TIMER PERIOD
88                 --
89                 dbms_output.put_line('test flag = ' || test_flag);
90                 SELECT CLASS_ID
91                 INTO V_CLASS_ID
92                 FROM SECTIONS_CLASSES
93                 WHERE SECTION_ID = LECTURER_SECTIONS_REC.SECTION_ID
94                 AND PERIOD_ID = V_PRD_ID
95                 AND ROWNUM = 1
96                 ORDER BY ID ASC;
97
98                 -- IF FOUND THEN
99                 dbms_output.put_line('to be inserted - exit now');
100             END;
101             -- EXIT SECTIONS_PERIODS_loop;
102
103             EXCEPTION
104                 WHEN NO_DATA_FOUND THEN
105                     DBMS_OUTPUT.PUT_LINE('N D F ' || LECT_CRS_REC.ID || ', class = ' || V_CLASS_ID ||
106                     'prd = ' || v_prd_id || ', lecturer id = ' || LECT_CRS_REC.LECTURER_ID );
107                     --CONTINUE;
108                 WHEN OTHERS THEN
109                     DBMS_OUTPUT.PUT_LINE('OTHER');
110             END;
111             END LOOP;
112
113             /*
114             if test_flag = 18 then -- loop SECTIONS_CLASSES finished with no class
115             -- available for this period
116
117             UPDATE SECTIONS_PERIODS
118             SET TICK = 1
119             WHERE SECTION_ID = LECTURER_SECTIONS_REC.SECTION_ID
120             AND PERIOD_ID IN ( SELECT PERIOD_ID FROM LECTURERS_PERIODS
121                             WHERE LECTURER_ID = LECT_CRS_REC.LECTURER_ID
122                             AND TICK = 0)
123             AND TICK = 0
124             AND ROWNUM = 1;
125
126             continue;
127             end if;
128
129             -- END LOOP SECTIONS_PERIODS_loop;
130
131
132             -- Check the same grp with same crs
133             -- not to be inserted again in table lectures
134             SELECT COUNT(*)
135             INTO V_CNT FROM LECTURES
136             WHERE SECTION_ID = LECTURER_SECTIONS_REC.SECTION_ID
137             AND COURSE_ID = LECT_CRS_REC.COURSE_ID
138             AND LECTURE_TIME = 1;
```

Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.INSERT_SECTIONS)]

File Edit Search Editor Session Database Debug View Utilities Window Help

ZAG_SCHED@XE

Schema Browser

INSERT_SECTIONS: Created: 28/06/2021 15:35:40 Last DDL: 28/06/2021 15:35:40 Status: Valid

Source Arguments Deps (Used) Deps (Used By) Errors Grants Synonyms Auditing

```
94 ( SELECT CLASS_ID FROM CLASSES_PERIODS -- LECTURES CLASSES WITH PERIODS
95 WHERE PERIOD_ID = V_PRD_ID
96 AND ROWNUM = 1
97 ORDER BY ID ASC;
98 -- IF FOUND THEN
99 dbms_output.put_line('to be inserted - exit now');
100 EXIT;
101 -- EXIT SECTIONS_PERIODS_loop;
102
103 EXCEPTION
104     WHEN NO_DATA_FOUND THEN
105         DBMS_OUTPUT.PUT_LINE('N D F ' || LECT_CRS_REC.ID || ', class = ' || V_CLASS_ID ||
106         'prd = ' || v_prd_id || ', lecturer id = ' || LECT_CRS_REC.LECTURER_ID );
107         --CONTINUE;
108     WHEN OTHERS THEN
109         DBMS_OUTPUT.PUT_LINE('OTHER');
110     END;
111     END LOOP;
112
113     /*
114     if test_flag = 18 then -- loop SECTIONS_CLASSES finished with no class
115     -- available for this period
116
117     UPDATE SECTIONS_PERIODS
118     SET TICK = 1
119     WHERE SECTION_ID = LECTURER_SECTIONS_REC.SECTION_ID
120     AND PERIOD_ID IN ( SELECT PERIOD_ID FROM LECTURERS_PERIODS
121                     WHERE LECTURER_ID = LECT_CRS_REC.LECTURER_ID
122                     AND TICK = 0)
123     AND TICK = 0
124     AND ROWNUM = 1;
125
126     continue;
127     end if;
128
129     -- END LOOP SECTIONS_PERIODS_loop;
130
131
132     -- Check the same grp with same crs
133     -- not to be inserted again in table lectures
134     SELECT COUNT(*)
135     INTO V_CNT FROM LECTURES
136     WHERE SECTION_ID = LECTURER_SECTIONS_REC.SECTION_ID
137     AND COURSE_ID = LECT_CRS_REC.COURSE_ID
138     AND LECTURE_TIME = 1;
```

```

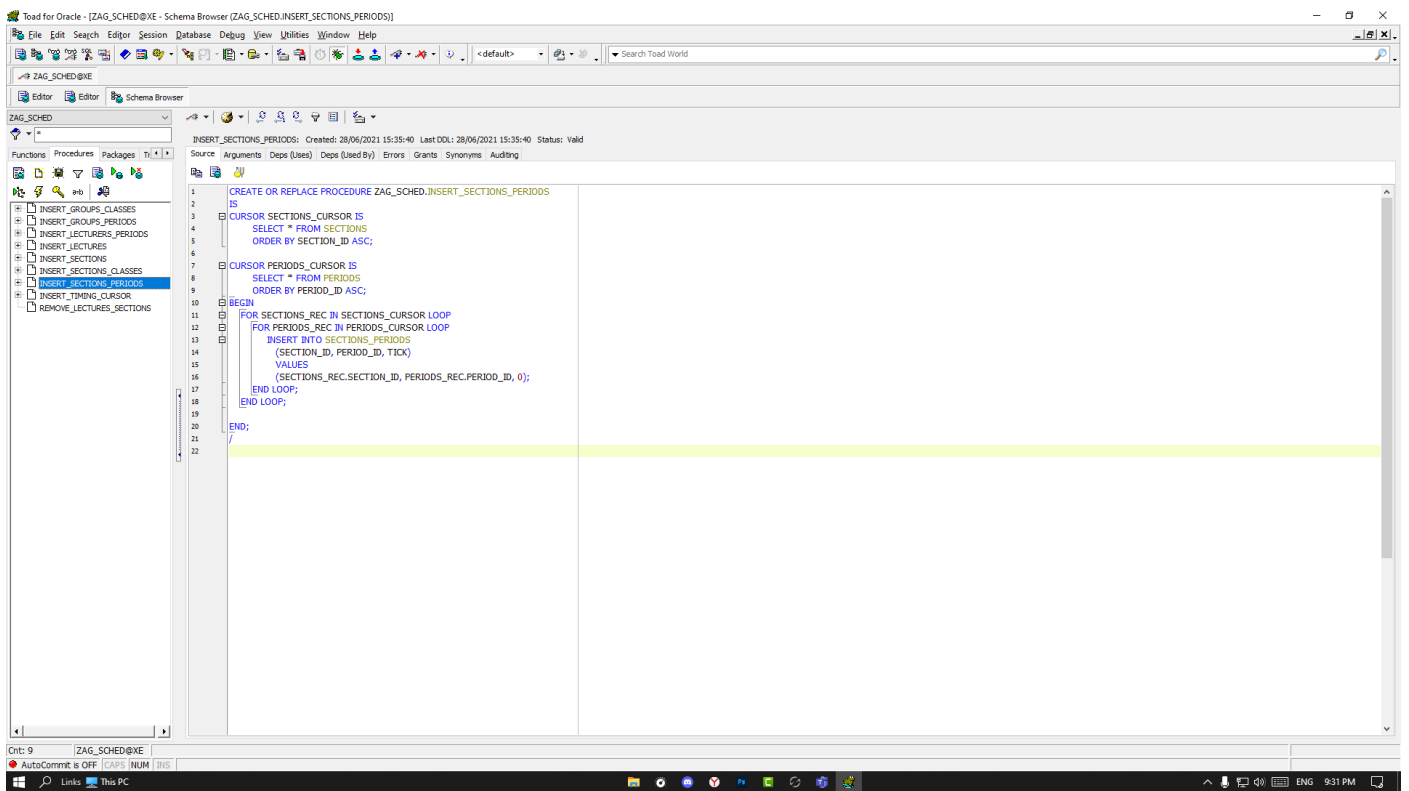
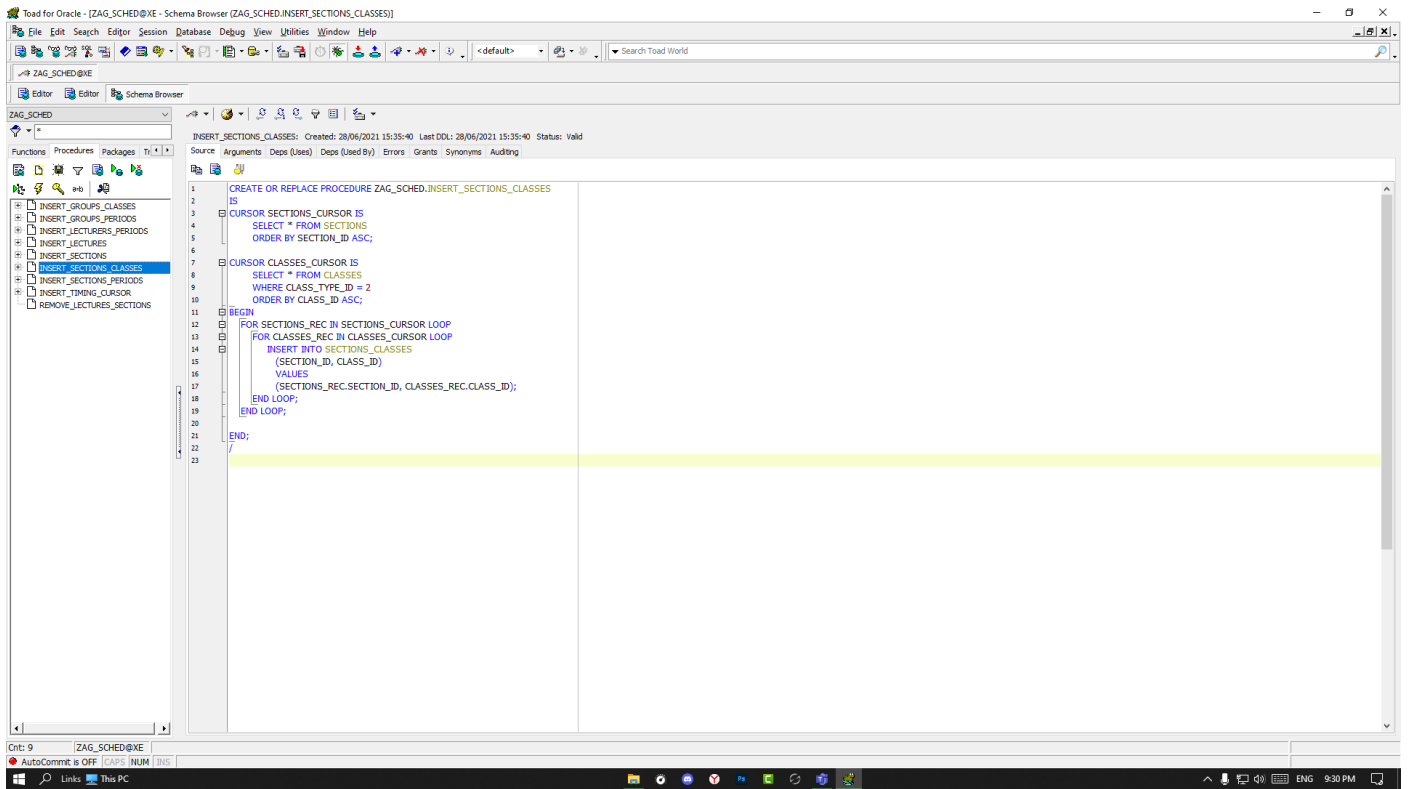
Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.INSERT_SECTIONS)]
File Edit Search Editor Session Database Debug View Utilities Window Help
ZAG_SCHED@XE
Schema Browser
ZAG_SCHED
INSERT_SECTIONS: Created: 28/06/2021 15:35:40 Last DDL: 28/06/2021 15:35:40 Status: Valid
Source Arguments Deps (Used) Deps (Used By) Errors Grants Synonyms Auditing
Functions Procedures Packages
INSERT_GROUPS_CLASSES
INSERT_GROUPS_PERIODS
INSERT_LECTURES_PERIODS
INSERT_LECTURES
INSERT_SECTIONS
INSERT_SECTIONS_CLASSES
INSERT_SECTIONS_PERIODS
INSERT_TIMING_CURSOR
REMOVE_LECTURES_SECTIONS
139
140
141 IF V_CNT = 1 THEN
142     V_CNT := 0;
143     DBMS_OUTPUT.PUT_LINE('V_CNT = 0' || LECT_CRS_REC_ID);
144     CONTINUE;
145 END IF;
146
147 -- DBMS_OUTPUT.PUT_LINE(LECT_CRS_REC.LECTURER_ID);
148
149 -- INSERT INTO LECTURES ( 1 row per loop )
150
151 INSERT INTO LECTURES
152 (LECTURE_TYPE_ID, SECTION_ID,
153  COURSE_ID, LECTURER_ID)
154 VALUES
155 (2, LECTURER_SECTIONS_REC.SECTION_ID, LECT_CRS_REC.COURSE_ID,
156  LECT_CRS_REC.LECTURER_ID);
157
158
159 SELECT COUNT(*)
160 INTO v_CNT_FAILED
161 FROM CLASSES_PERIODS
162 WHERE PERIOD_ID = V_PRD_ID
163 AND CLASS_ID = V_CLASS_ID;
164
165 INSERT INTO CLASSES_PERIODS -- ( 1 row per loop )
166 (PERIOD_ID, CLASS_ID, LECTURE_ID)
167 VALUES
168 (V_PRD_ID, V_CLASS_ID, LECTURES_SEQ.CURRVAL);
169
170
171 DBMS_OUTPUT.PUT_LINE('Inserted SEQ = ' ||
172 CLASSES_PERIODS_SEQ.CURRVAL || ', Section = ' ||
173 LECTURER_SECTIONS_REC.SECTION_ID
174 || ' lec crs id = ' || LECT_CRS_REC.COURSE_ID
175 || ', class = ' || V_CLASS_ID ||
176 ' prd = ' || V_PRD_ID || ', lecturer id = ' || LECT_CRS_REC.LECTURER_ID );
177
178 IF V_CNT_FAILED > 0 THEN
179     DBMS_OUTPUT.PUT_LINE('Failed = ' || CLASSES_PERIODS_SEQ.CURRVAL);
180 end if;
181 -- THIS PERIOD WITH THIS GROUP IS TAKEN NOW
182
183 UPDATE SECTIONS_PERIODS
184 SET TICK = 1
185 WHERE SECTION_ID = V_SECTION_ID
186 AND PERIOD_ID = V_PRD_ID;
187
188 -- DBMS_OUTPUT.PUT_LINE('ISSUE _ LECTURER = ' || LECT_CRS_REC.LECTURER_ID || ', PRD = ' ||
189 V_PRD_ID);
190
191
192 -- THIS PERIOD WITH THIS LECTURER IS TAKEN
193 UPDATE LECTURERS_PERIODS
194 SET TICK = 1
195 WHERE LECTURER_ID = LECT_CRS_REC.LECTURER_ID
196 AND PERIOD_ID = V_PRD_ID;
197
198 DBMS_OUTPUT.PUT_LINE('ISSUE _ LECTURER = ' || LECT_CRS_REC.LECTURER_ID || ', PRD = ' ||
199 V_PRD_ID);
200
201 END LOOP; -- END LECTURER_SECTIONS
202
203 END LOOP; -- END LECTURES_COURSES
204
205 DBMS_OUTPUT.PUT_LINE('V_CNT_LLECT_CRS ' || V_CNT_LLECT_CRS);
206 V_CNT_LLECT_CRS := 0;
207 DBMS_OUTPUT.PUT_LINE(' ');
208 -- END LOOP; -- END GROUPS
209 -- END LOOP; -- END DAYS
210 END LOOP; -- END DIVISIONS_CURSOR
211
212 commit;
213 END;
214
215

```

```

Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.INSERT_SECTIONS)]
File Edit Search Editor Session Database Debug View Utilities Window Help
ZAG_SCHED@XE
Schema Browser
ZAG_SCHED
INSERT_SECTIONS: Created: 28/06/2021 15:35:40 Last DDL: 28/06/2021 15:35:40 Status: Valid
Source Arguments Deps (Used) Deps (Used By) Errors Grants Synonyms Auditing
Functions Procedures Packages
INSERT_GROUPS_CLASSES
INSERT_GROUPS_PERIODS
INSERT_LECTURES_PERIODS
INSERT_LECTURES
INSERT_SECTIONS
INSERT_SECTIONS_CLASSES
INSERT_SECTIONS_PERIODS
INSERT_TIMING_CURSOR
REMOVE_LECTURES_SECTIONS
178
179 IF V_CNT_FAILED > 0 THEN
180     DBMS_OUTPUT.PUT_LINE('Failed = ' || CLASSES_PERIODS_SEQ.CURRVAL);
181 end if;
182 -- THIS PERIOD WITH THIS GROUP IS TAKEN NOW
183
184 UPDATE SECTIONS_PERIODS
185 SET TICK = 1
186 WHERE SECTION_ID = V_SECTION_ID
187 AND PERIOD_ID = V_PRD_ID;
188
189 -- DBMS_OUTPUT.PUT_LINE('ISSUE _ LECTURER = ' || LECT_CRS_REC.LECTURER_ID || ', PRD = ' ||
190 V_PRD_ID);
191
192
193 -- THIS PERIOD WITH THIS LECTURER IS TAKEN
194 UPDATE LECTURERS_PERIODS
195 SET TICK = 1
196 WHERE LECTURER_ID = LECT_CRS_REC.LECTURER_ID
197 AND PERIOD_ID = V_PRD_ID;
198
199 DBMS_OUTPUT.PUT_LINE('ISSUE _ LECTURER = ' || LECT_CRS_REC.LECTURER_ID || ', PRD = ' ||
200 V_PRD_ID);
201
202 END LOOP; -- END LECTURER_SECTIONS
203
204 END LOOP; -- END LECTURES_COURSES
205
206 DBMS_OUTPUT.PUT_LINE('V_CNT_LLECT_CRS ' || V_CNT_LLECT_CRS);
207 V_CNT_LLECT_CRS := 0;
208 DBMS_OUTPUT.PUT_LINE(' ');
209 -- END LOOP; -- END GROUPS
210 -- END LOOP; -- END DAYS
211 END LOOP; -- END DIVISIONS_CURSOR
212
213 commit;
214 END;
215

```



Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.INSERT_TIMING_CURSOR)]

File Edit Search Editor Session Database Debug View Utilities Window Help

Search Toad World

ZAG_SCHED@XE

Editor Editor Schema Browser

ZAG_SCHED

INSERT_TIMING_CURSOR: Created: 28/06/2021 15:35:40 Last DDL: 28/06/2021 15:35:40 Status: Valid

Source Arguments Deps (Used) Deps (Used By) Errors Grants Synonyms Auditing

Functions Procedures Packages

INSERT_GROUPS_CLASSES
INSERT_GROUPS_PERIODS
INSERT_LECTURES_PERIODS
INSERT_LECTURES
INSERT_SECTIONS
INSERT_SECTIONS_CLASSES
INSERT_SECTIONS_PERIODS
INSERT_TIMING_CURSOR
REMOVE_LECTURES_SECTIONS

```
1 CREATE OR REPLACE PROCEDURE ZAG_SCHED.INSERT_TIMING_CURSOR
2 IS
3 CURSOR PERIODS_TIMING_CURSOR IS
4 SELECT * FROM PERIODS_TIMING;
5
6 BEGIN
7
8 FOR I IN 1..5 LOOP
9
10 INSERT INTO PERIODS_TIMING
11 (TIME_TEXT)
12 VALUES
13 ('8.30');
14
15 INSERT INTO PERIODS_TIMING
16 (TIME_TEXT)
17 VALUES
18 ('10.00');
19
20 INSERT INTO PERIODS_TIMING
21 (TIME_TEXT)
22 VALUES
23 ('10.15');
24
25 INSERT INTO PERIODS_TIMING
26 (TIME_TEXT)
27 VALUES
28 ('11.45');
29
30 INSERT INTO PERIODS_TIMING
31 (TIME_TEXT)
32 VALUES
33 ('12.00');
34
35 INSERT INTO PERIODS_TIMING
36 (TIME_TEXT)
37 VALUES
38 ('1.30');
39
40 INSERT INTO PERIODS_TIMING
41 (TIME_TEXT)
42 VALUES
43 ('1.45');
44
45 INSERT INTO PERIODS_TIMING
```

Obj: 9 ZAG_SCHED@XE

AutoCommit is OFF [CAPS NUM] [TNS]

Links This PC [TNS]

ENG 9:31 PM

Toad for Oracle - [ZAG_SCHED@XE - Schema Browser (ZAG_SCHED.REMOVE_LECTURES_SECTIONS)]

File Edit Search Editor Session Database Debug View Utilities Window Help

Search Toad World

ZAG_SCHED@XE

Editor Editor Schema Browser

ZAG_SCHED

REMOVE_LECTURES_SECTIONS: Created: 28/06/2021 15:35:40 Last DDL: 28/06/2021 15:35:40 Status: Valid

Source Arguments Deps (Used) Deps (Used By) Errors Grants Synonyms Auditing

Functions Procedures Packages

INSERT_GROUPS_CLASSES
INSERT_GROUPS_PERIODS
INSERT_LECTURES_PERIODS
INSERT_LECTURES
INSERT_SECTIONS
INSERT_SECTIONS_CLASSES
INSERT_SECTIONS_PERIODS
INSERT_TIMING_CURSOR
REMOVE_LECTURES_SECTIONS

```
1 CREATE OR REPLACE PROCEDURE ZAG_SCHED.REMOVE_LECTURES_SECTIONS
2 IS
3 BEGIN
4 DELETE CLASSES_PERIODS;
5 DELETE LECTURES;
6
7 UPDATE GROUPS_PERIODS
8 SET TJCK = 0;
9
10 UPDATE LECTURERS_PERIODS
11 SET TJCK = 0;
12
13 UPDATE SECTIONS_PERIODS
14 SET TJCK = 0;
15
16 commit;
17
18 END;
19
20 /
```

Obj: 9 ZAG_SCHED@XE

AutoCommit is OFF [CAPS NUM] [TNS]

Links This PC [TNS]

ENG 9:31 PM

Chapter Four

User Interface

4 GUI Design

4.1 Login Screen

- Login Screen is the first screen in our project, it just start with button and text boxes.



Zag Sched - شاشة الدخول

دخول البرنامج

إسم المستخدم

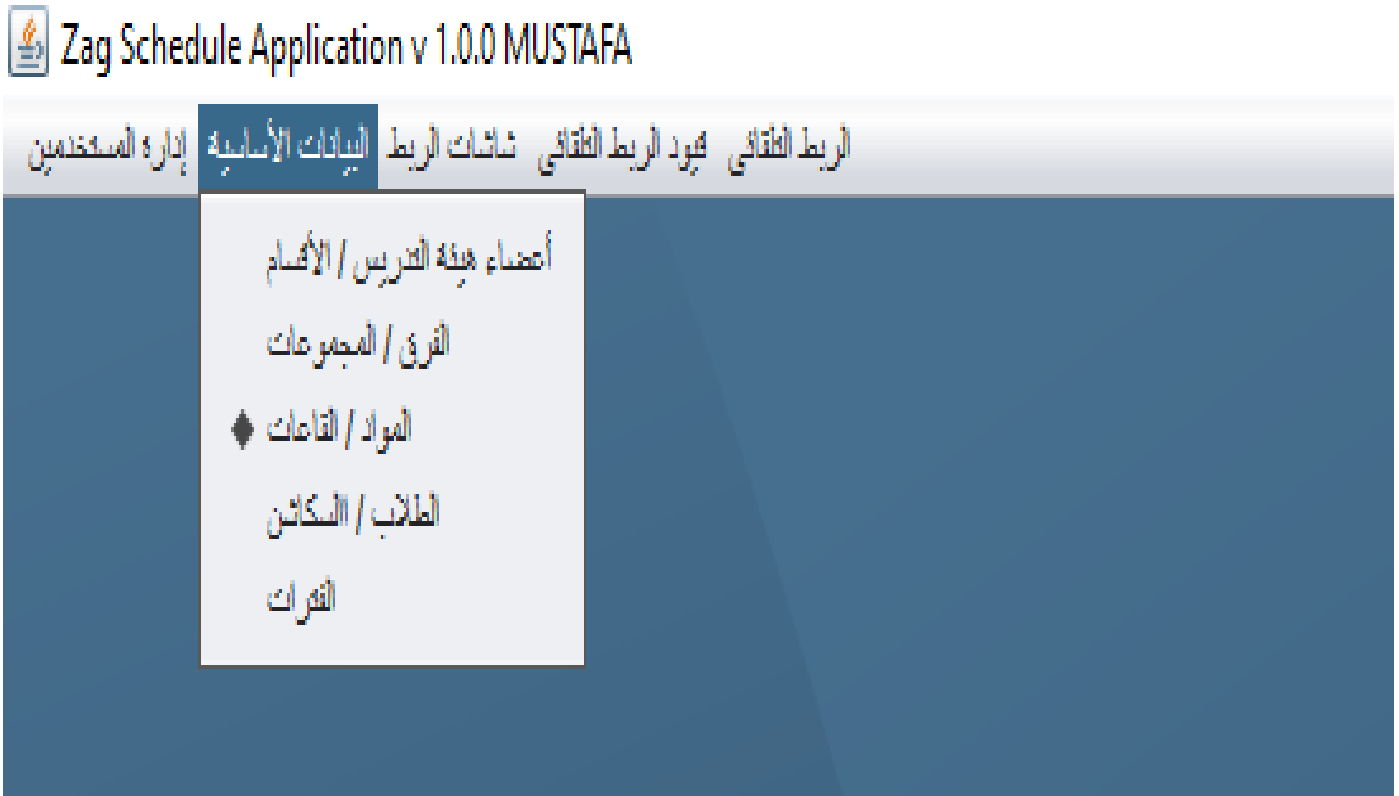
كلمة السر

دخول



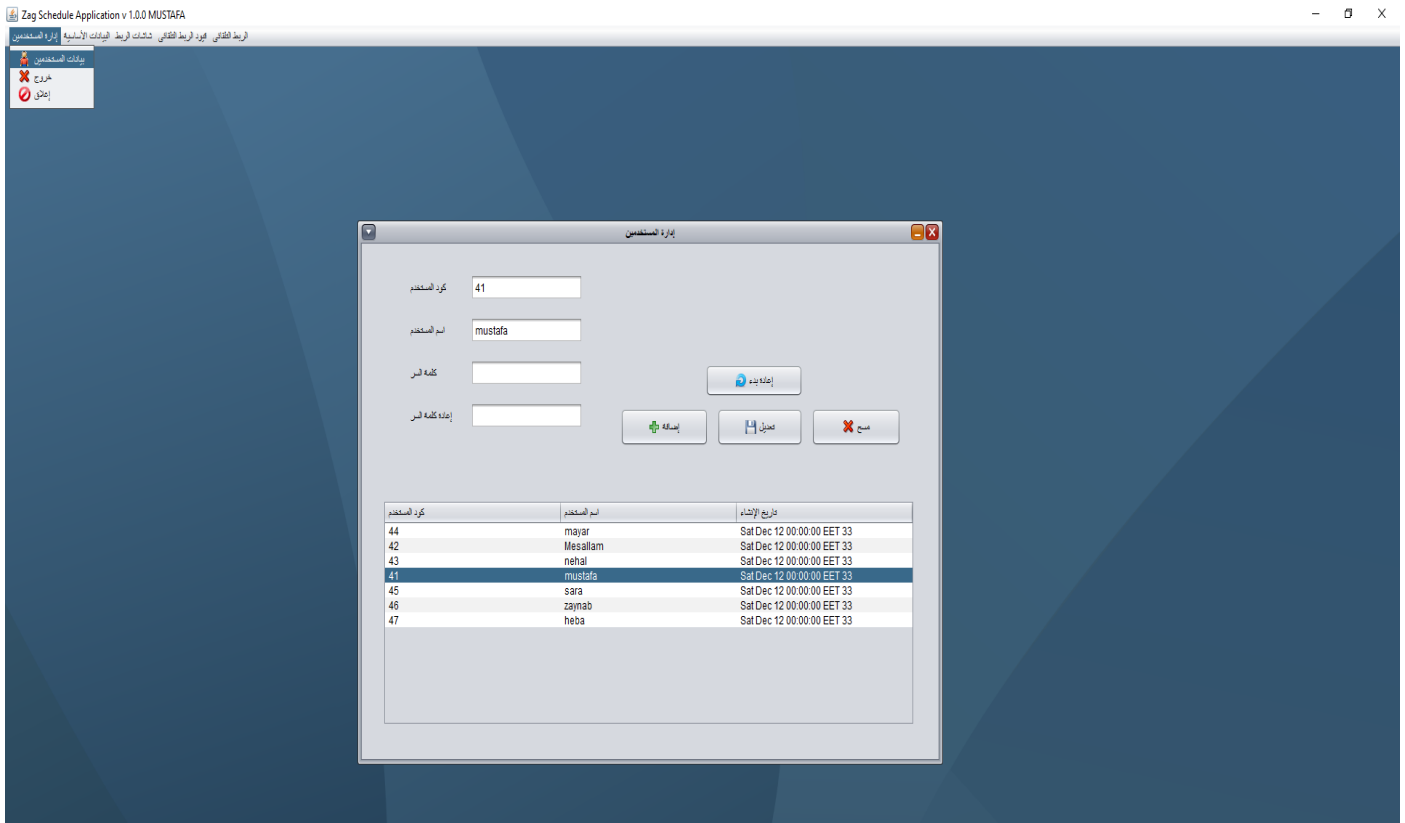
4.2 Main Screen

- Main screen you can browse the app throw it, contains drop menus.



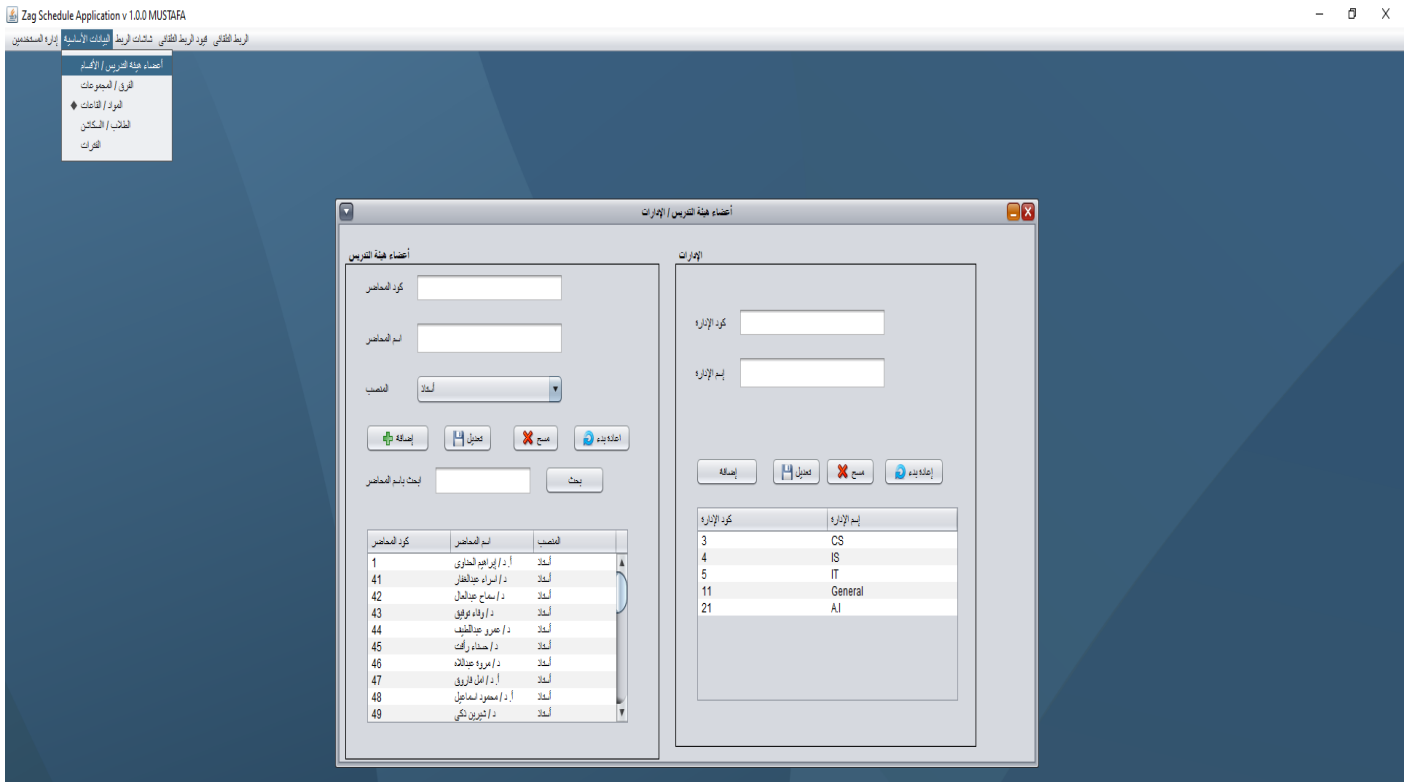
4.3 Users Screen

- Users Screen, Contains users Id , Users names



4.4 Teaching Stuff and Departments Screen

- Teaching Stuff Screen Contains, Doctors Information and Departments.



4.5 Divisions and Groups

- Divisions and Groups screen contains, Divisions information's and Groups numbers.

Zag Schedule Application v 1.0.0 MUSTAFA

الرابط الفعلي: يعود الرابط الفعلي: بيانات الرابط: أدوات الإحصائية: إدارة المستخدمين

أعضاء هيئة التدريس / الأقسام
الفرق / المجموعات
المواد / القاعات
المحاضرات / المحاضرات

الفرق / المجموعات

الفرق

كود الفرق:

اسم الفرق:

الإدارة:

كود الفرق	اسم الفرق	الإدارة
11	الفرقة الأولى	General
12	الفرقة الثانية	General
13	الفرقة الثالثة	General
14	الفرقة الرابعة CS	CS
15	الفرقة الرابعة IS	IS
16	الفرقة الرابعة IT	IT
17	الفرقة الرابعة AI	AI

المجموعات

كود المجموعة:

اسم المجموعة:

الفرقة:

كود المجموعة	اسم المجموعة	الفرقة
6	المجموعة الأولى - فرقة أولى	الفرقة الأولى
7	المجموعة الثانية - فرقة أولى	الفرقة الأولى
8	المجموعة الثالثة - فرقة أولى	الفرقة الأولى
9	المجموعة الرابعة - فرقة أولى	الفرقة الأولى
10	المجموعة الخامسة - فرقة أولى	الفرقة الأولى
22	المجموعة السادسة - فرقة أولى	الفرقة الأولى
23	المجموعة السابعة - فرقة أولى	الفرقة الأولى
25	المجموعة الثامنة - فرقة أولى	الفرقة الأولى
27	المجموعة التاسعة - فرقة أولى	الفرقة الأولى
28	المجموعة العاشرة - فرقة أولى	الفرقة الأولى



4.6 Classes and Courses

- Classes and Courses Screen contains, Classes information's and Courses Names.

Zag Schedule Application v 1.0.0 MUSTAFA

الرابط التلقائي: كود الربط التلقائي: بيانات الربط الأساسية: إدارة المستخدمين

أعضاء هيئة التدريس / الأقسام
الفرق / المجموعات
المواد / الفئات
الطلاب / الكائنات
الفترات

الفئات

كود الفئة

اسم الفئة

نوع الفئة: مدرج محاضرات

إضافة تعديل مسح إعادة بدء

كود الفئة	اسم الفئة	نوع الفئة
21	مدرج 1	مدرج محاضرات
22	مدرج 2	مدرج محاضرات
23	مدرج 3	مدرج محاضرات
24	مدرج 4	مدرج محاضرات
41	معمل 1	معمل
42	معمل 2	معمل
43	معمل 3	معمل
44	معمل 4	معمل
45	معمل 5	معمل
46	معمل 6	معمل

المواد

كود المادة

اسم المادة

إضافة تعديل مسح إعادة بدء

كود المادة	اسم المادة
1	برمجة 2
2	رياضة 2
4	حقوق الإنسان ومكافحة الفساد
5	بحوث مصريات
6	تراثنا مصرية
7	تصميم منطقي
21	القضايا الاجتماعية والأخلاقية والمهنية ف...
61	وسائط متعددة
62	تحليل وتصميم النظم
63	نظم خبيرة



4.7 Students and Sections

- Students and sections Screen contains, students information's and Sections information's .

Zag Schedule Application v 1.0.0 MUSTAFA

الرابط الشفائي: [يود الربط الشفائي](#) - [جائت الربط](#) - [البيانات الأساسية](#) - [إدارة المستخدمين](#)

أقسام هيئة التدريس / الأقسام
الفرق / المجموعات
المواد / القاعات
الطلاب / السكتن
الهيئات

الطلاب / السكتن

الطلاب

كود الطالب:

اسم الطالب:

الفرقة:

المجموعة:

السكتن:

القسم:

الاسم	السكتن	المجموعة	الفرقة	اسم الطالب	كود الطالب
-------	--------	----------	--------	------------	------------

السكتن

كود السكتن:

اسم السكتن:

المجموعة:

المجموعة	اسم السكتن	كود السكتن
المجموعة الأولى - فرقة دالة	سكتن 1	1
المجموعة الأولى - فرقة دالة	سكتن 2	42
المجموعة الأولى - فرقة دالة	سكتن 3	43
المجموعة الأولى - فرقة دالة	سكتن 4	45
المجموعة الأولى - فرقة دالة	سكتن 5	46
المجموعة الأولى - فرقة دالة	سكتن 6	47
المجموعة الأولى - فرقة دالة	سكتن 7	48
المجموعة الأولى - فرقة دالة	سكتن 8	49



4.8 Periods

- Periods Screen contains, Name and Periods Information's.

Zag Schedule Application v 1.0.0 MUSTAFA

إدارة المستخدمين | الربط التلقائي | فترات الربط | بيانات الأساسية

أعضاء هيئة التدريس / الأقسام
الفرق / المجموعات
المواد / القاعات
الطلاب / السكاكين
الفترات

الفترات

كود الفترة:

اسم الفترة:

اليوم:

من ساعة:

إلى ساعة:

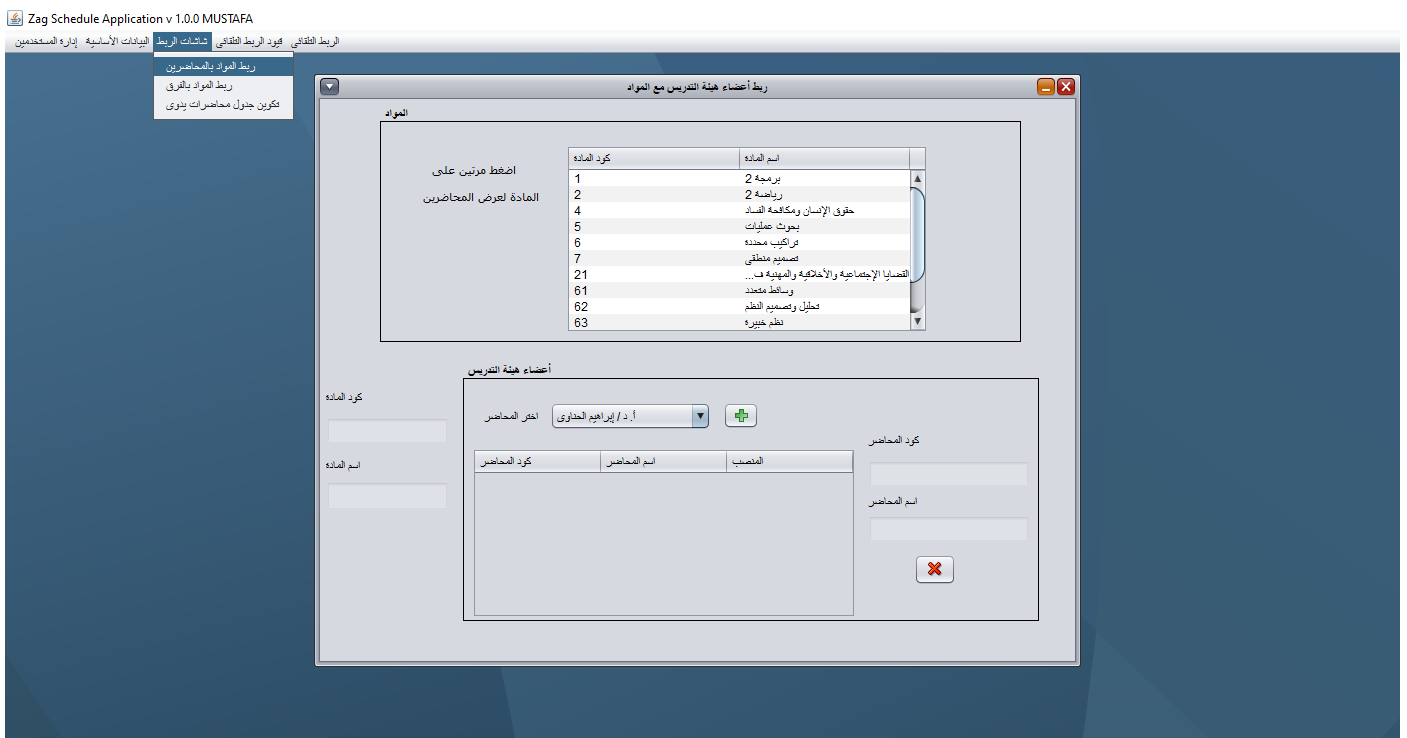
إضافة مسح

كود الفترة	اسم الفترة	اليوم	من ساعة	إلى ساعة
1	8.30 : 10...	السبت	8.30	10.00
4	1.45 : 3....	السبت	1.45	3.15
5	8.30 : 10...	الأحد	8.30	10.00
6	10.15 : 1...	الأحد	10.15	11.45
7	12.00 : 1...	الأحد	12.00	1.30
8	1.45 : 3....	الأحد	1.45	3.15
9	8.30 : 10...	الاثنين	8.30	10.00
10	10.15 : 1...	الاثنين	10.15	11.45
11	12.00 : 1...	الاثنين	12.00	1.30
12	1.45 : 3....	الاثنين	1.45	3.15



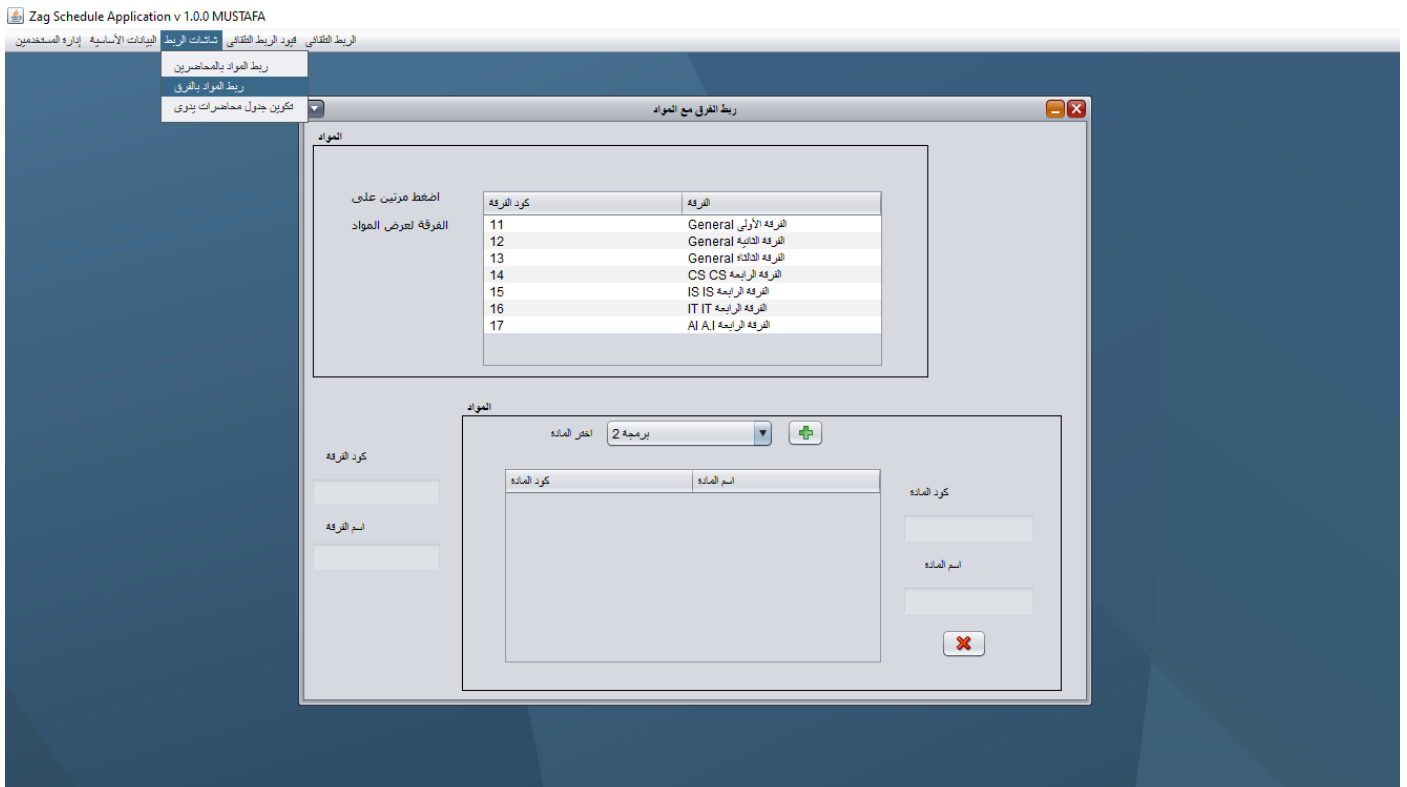
4.9 Lecturers And Courses

- Lecturers and courses Screen contains, Doctors information's and courses names.



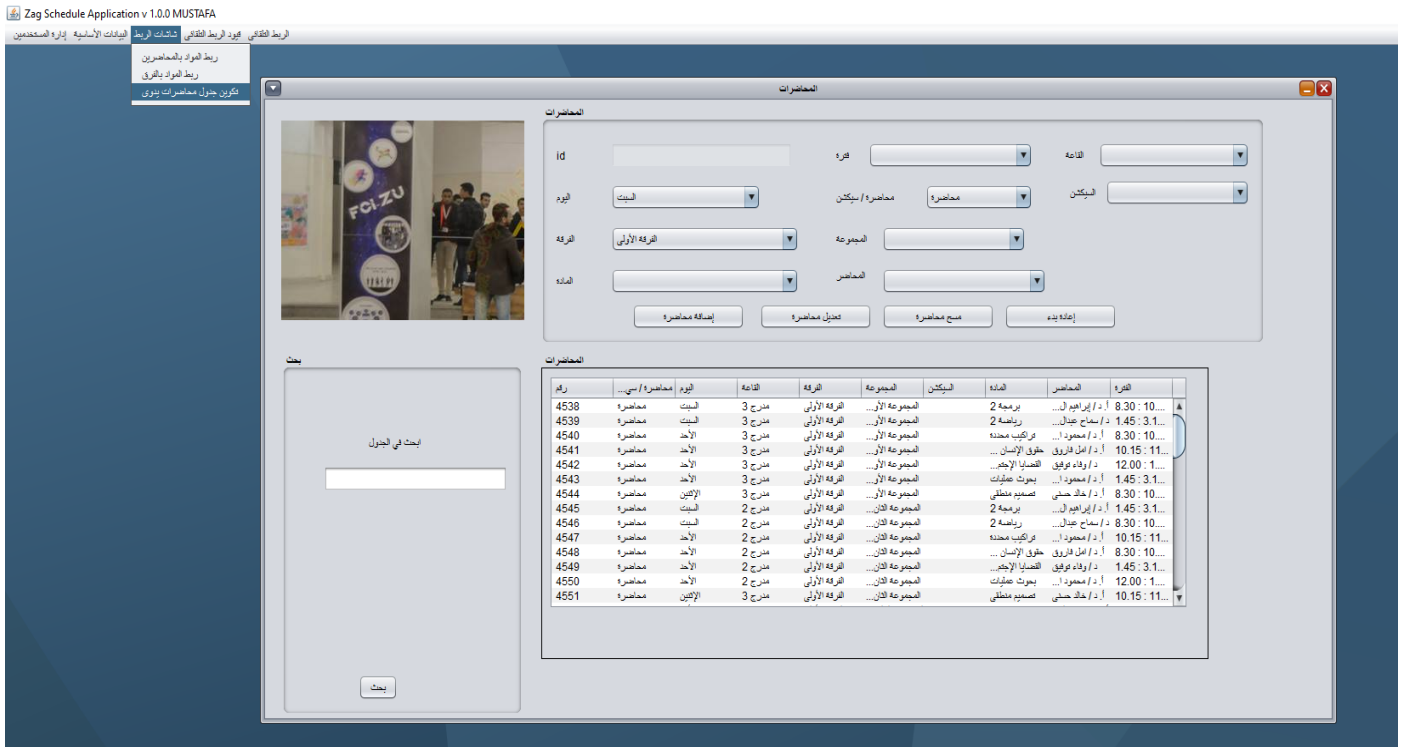
4.10 Divisions And Courses

- Divisions and Courses Screen contains, Divisions names and courses names.



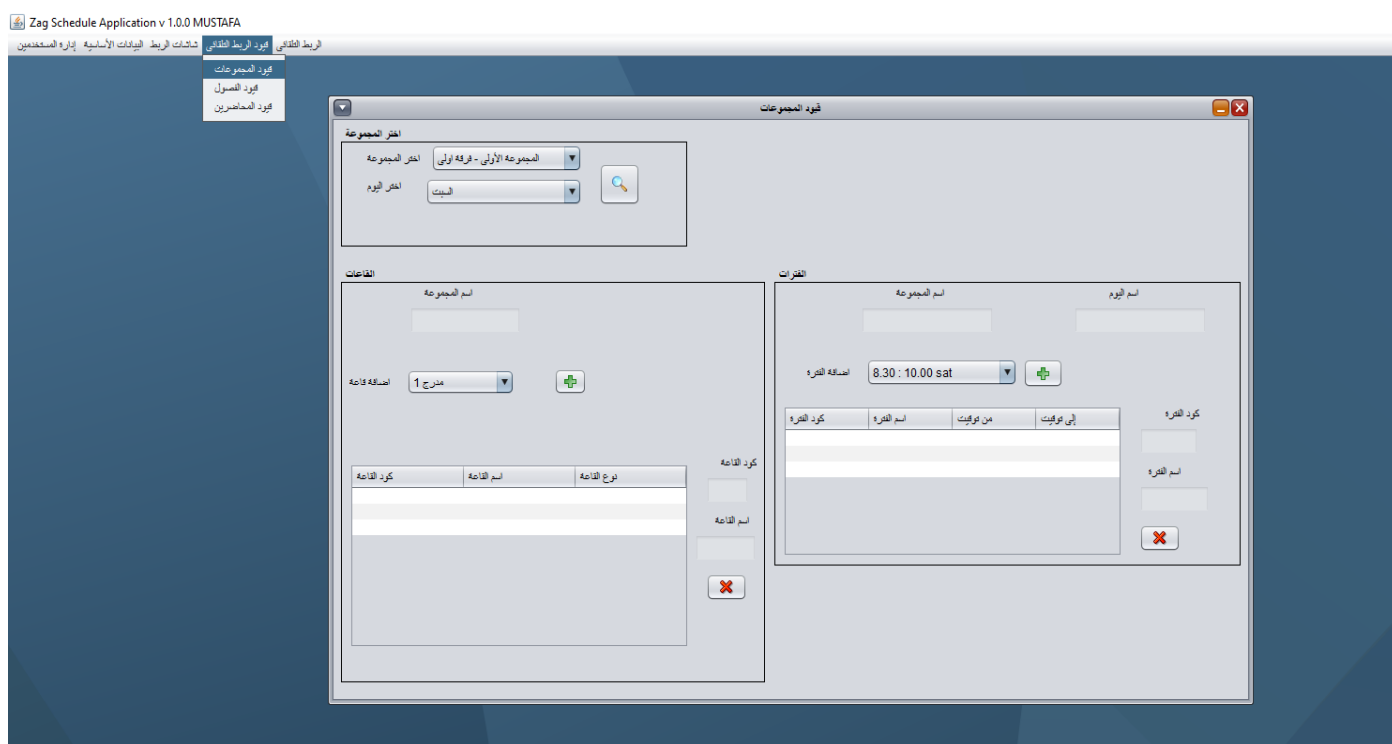
4.11 Create Schedule Manually

- Create schedule Screen contains, A tool used for Create Schedules Manually.



4.12 Groups Constrains

- Groups Constrains Screen contains, Classes and Periods Constrains.



4.13 Sections Constrains

- Sections Constrains Screen contains, Sections and Periods Constrains.

Zag Schedule Application v 1.0.0 MUSTAFA

الرابط التفتاحي: شارات الربط الأساسية: إدارة المستخدمين

إيراد الميعونات
إيراد الفصول
إيراد المحاضرين

قيود الفصول

اختر الفصل:

اختر اليوم:

القيود

اسم الفصل:

اسم الفترة:

كود القاعة:

كود القاعة	اسم القاعة	نوع القاعة

اسم القاعة:

اسم الفصل:

اسم اليوم:

اسم الفترة:

كود الفترة:

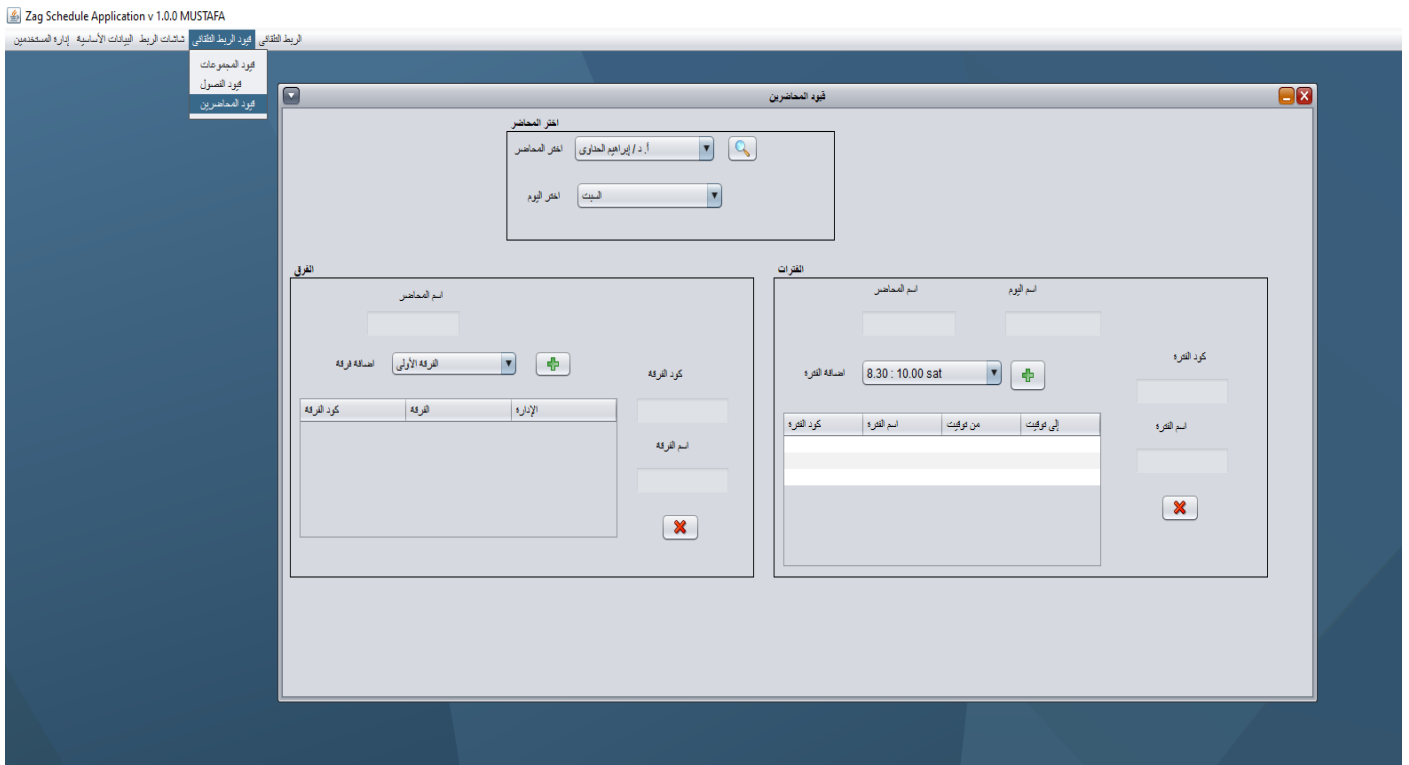
كود الفترة	اسم الفترة	من توقيت	إلى توقيت

اسم الفترة:



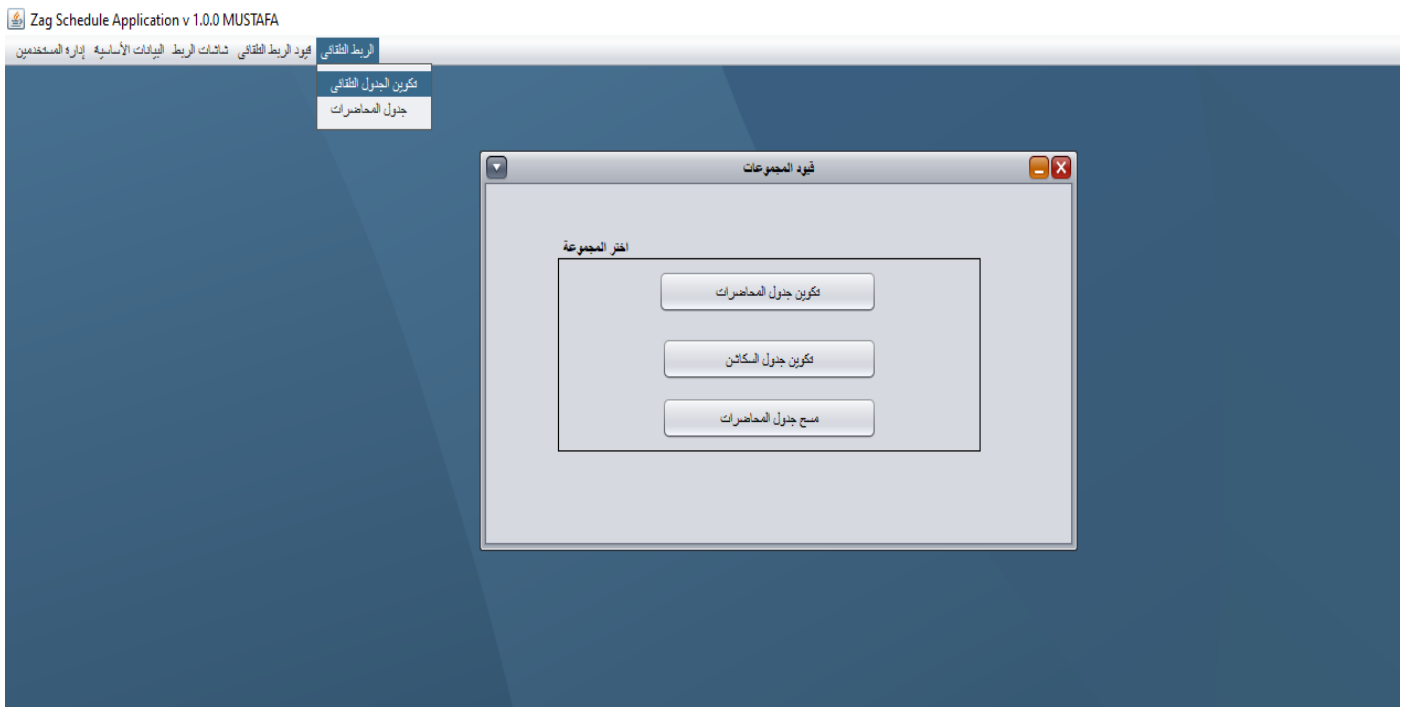
4.14 Lecturers Constrains

- Lecturers Constrains Screen contains, Groups and Periods Constrains.



4.15 Generating Options Screen

- Generating Options Screen contains, Lecture Schedule Generate and Sections Schedule Generate and Delete Schedules.



4.16 Generated Schedule

- Generated schedule Screen Contains, the Schedule and options to modify it.

The Most Important Screen

[illegible]

Chapter Five

Conclusions and Future work

5.1 Conclusions

Our application is built for helping college and university on a semester problem and finding the fastest solution to the problems by using your desktop application, push you some tapes to edit or insert data and by click on create table you will got the table without any conflict.

- When user run the application, asked for user name and password to log in.
- After log in, will show some tapes, help for creating table.
- The last tape has:
 1. Create lecture table
 2. Create section table
 3. Delete table

When user click on create table, system run and generate table based on the data and the conflict that will enter before.

5.2 Future Work

- Add an Expert system :

In the future we will add an expert system that will provide help for user and make the enter data and constraints easy and fast, this expert system will depend on algorithms and filtering process to filter and sort the constraints. Expert system will use an Artificial Intelligence Technology to learn from the previous entered constraints and provide user with more Stable replays.

