



Change Request Tracker

The Change Management System

CIS 220 Database Concepts

G2 Group Study

Table Of Contents

Table Of Contents	1
Team Members	2
Our Mission	2
Objectives	3
Intended For	3
Scope	3
User Requirements	3
Client	4
Current System	4
Problems In Current Systems	4
Use Case Scenarios	5
Basic Flow of Events	5
Feasibility	6
Expectations	6
Limitations	6
Normalization	7
Initial Conditions	7
First Normal Form	7
Second Normal Form and Third Normal Form	7
Data Requirements	8
Individual Tables	8
Users	8
Departments	8
DepartmentRequestTypes	9
ChangeRequests	9
StakeholderRequests	9
RequestNotes	10
RequestType	10
Grand Totals for Data Tables	10
Yearly Storage Cost Estimates	11
Entity Relationship Diagram	12
Business Rules	13

Team Members

Jordan Aceto, Sr. Programmer

Jordan is committed to providing error free code. He understands what it takes to make changes to Production code with no impact to end-users.

Patrick Gorst, Database Administrator

Patrick is determined to provide best practices to company change management. When the data is exact, the changes are exact.

Our Mission

Our mission is to provide a modest change request management database system, suitable for small or medium sized businesses.

Objectives

Intended For

Our database is intended for any small (100 users) to medium (400-1000 users) sized business which needs to streamline and standardize its change request management.

Scope

- Store user change request, this can relate to either production, dev, or general active environments
- Archive change request for later look-back and or ad-hoc searches
- Allow leadership to approve or deny changes
- Allow managers to track changes from department
- Notify and keep stakeholders aware of assigned change request
- Allow managers to track a change request progress

User Requirements

Users of this software are employees and managers in a small or medium sized business, where changes to employee's software or data access are requested and carried out. A change request is any internal change, examples include updating the intranet, adding drop down menus or GUI elements, changing access to reports, requesting updates to specific software, etc.

Users of this software can expect a more streamlined process for generating, completing, and tracking changes. This easy to use and intuitive change management software leads to less downtime and increased team cohesion.

Users of the software submit change requests. These requests are then approved by management and a due date is set. The request is then assigned to a stakeholder. As the stakeholder makes progress updates are sent to the user. When the request is completed, notifications are sent to the user and manager. Upon completion, the request is archived for future look-back.

Client

Current System

This database system is to replace any current systems (and non-system) of the following:

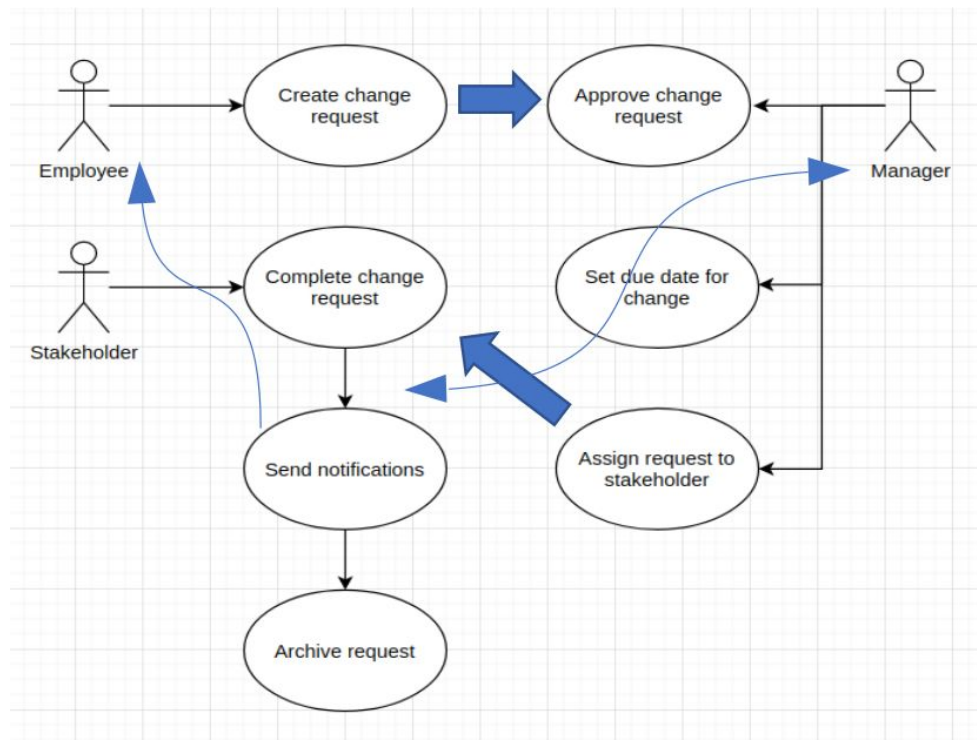
- IM
- Email
- Word of mouth
- Memo
- Any form of change notification, that has no process flow, and has resulted in system downtime or otherwise end-user impact

Problems In Current Systems

Although email seems reliable, send and receive traffic can be heavy. Missing a change request, or notification, occurs often. The proper stakeholders are often not included, or required resources are missing. The result is system downtime due to communication barriers. IM is easily missed or offline. Memos are ignored. People often forget. - Current systems typically lack change management and liability.

Use Case Scenarios

Example 1.0



Example 1.0 shows the fundamental flow, although complexity varies on demand, to target a variety of audiences modules are a possibility

Basic Flow of Events

- 1) User request a change
- 2) Change request is approved or denied by manager
- 3) Request is assigned a due date
- 4) Manager assigns to stakeholder(s) queue
- 5) Stakeholders add notes on progress and task or complete the whole request
- 6) Notifications are sent to user and manager
- 7) Change request is archived for later search

Feasibility

Expectations

The proposed change management system is well suited to be implemented by a database. Software requirements are modest, and no additional hardware should be required. The skills and knowledge required to complete the project will be learned during the course of the class.

Patrick brings a wealth of professional experience with databases and as Infrastructure Systems Administrator, an ideal background for completing this project.

Jordan has a keen understanding of programming best practices and analytics. Jordan's professional demeanor, work ethics, and problem solving skills, serve as the foundation of which this project is built.

Limitations

limitations of this database are dependent on the business in question, who will use this system. Not to be confused with a ticketing system, although similar, the idea of change request tracking is to focus on 'changes' that are required. If the business requires a more robust request and project management database\system, this system may not be enough, as in why it's intended for small to medium size environments. Otherwise, the performance and retention are **dependent on company resources**. We assume any potential partners are aware of ReChanges target audience. Work outside of the targeted scope (small-medium) cannot be accommodated and advise consideration for systems designed to support larger environments.

Normalization

Here we will break down the tables of this change tracking database, from first to third normal form. Refer to the basic flow of events and use case scenario for a visual presentation. (*Example 1.0*)

Initial Conditions

ChangeRequest (Request, Requester, Manager, Department, Stakeholders, RequestNotes, RequestArchive)

First Normal Form

ChangeRequest (*RequestID* (PK), RequestDescription, RequestDateStamp, RequestDueDate, RequestApprovaID, RequestApporvedDenied, UserID, FirstName, LastName, Email, Phone, Department, ManagerID, ManagerID, ManagerFirstName, ManagerLastName, ManagerDepartment, ManagerEmployeesID, DepartmentID, DepartmentName, DepartmentManagerID, StakeholdersID, StakeholdersDepartmentID, StakeholdersManagerID, RequestID, RequestNotesID, RequestNotesDecsription, RequestNotesDateStamp, OpenClosedComplete, RequestClosedNotes, RequestCompleteNotes, RequestTypeID, TopicLabel, Application)

Second Normal Form and Third Normal Form

ChangeRequest (*RequestID* (PK) => RequestDescription, RequestDateStamp, RequestDueDate, RequestApproved, UserID(FK), RequestTypeID(FK))

Users (*RequesterID* (PK) => FirstName, LastName, Email, Phone, DepartmentID(FK))

Departments (*DepartmentID*(PK) => DepartmentName, ManagerID(FK))

RequestNotes (*RequestNotesID*(PK) => RequestNotesDecsription, RequestNotesDateStamp, isOpen, isComplete, RequestClosedNotes, RequestCompleteNotes. RequestID(FK))

RequestType (RequestTypeID(PK) => TopicLable, Application)

StakeholderRequests(RequestID(PK,FK), StakeholderID(PK,FK))

DepartmentRequestTypes(DepartmentID(PK,FK), RequestTypeID(PK,FK))

Data Requirements

The data requirements for the tables in the system are outlined below. All data-size tables below are in units of bytes. Size needs are very modest and should not pose any issue. Data privacy is a concern; user email addresses are stored and some change requests may contain sensitive information.

Individual Tables

Users

Name	Data Type	Minimum Data Size	Maximum Data Size
<i>UserID</i>	INT	4	4
<i>DepartmentID</i>	INT	4	4
<i>FirstName</i>	VARCHAR(45)	1	46
<i>LastName</i>	VARCHAR(45)	1	46
<i>ReqEmail</i>	VARCHAR(45)	1	46
<i>Phone</i>	VARCHAR(45)	1	46
	Totals:	12	192

Departments

Name	Data Type	Minimum Data Size	Maximum Data Size
<i>DepartmentID</i>	INT	4	4
<i>ManagerID</i>	INT	4	4
<i>DepartmentName</i>	VARCHAR(45)	1	46
	Totals:	9	54

DepartmentRequestTypes

Name	Data Type	Minimum Data Size	Maximum Data Size
<i>DepartmentID</i>	INT	4	4
<i>RequestTypeID</i>	INT	4	4
	Totals:	8	8

ChangeRequests

Name	Data Type	Minimum Data Size	Maximum Data Size
<i>RequestID</i>	INT	4	4
<i>UserID</i>	INT	4	4
<i>RequestTypeID</i>	INT	4	4
<i>RequestDescription</i>	VARCHAR(45)	1	46
<i>RequestDateStamp</i>	DATETIME	8	8
<i>RequestDueDate</i>	DATETIME	8	8
<i>RequestApproved</i>	BOOLEAN	1	1
	Totals:	30	75

StakeholderRequests

Name	Data Type	Minimum Data Size	Maximum Data Size
<i>RequestID</i>	INT	4	4
<i>StakeholderID</i>	INT	4	4
	Totals:	8	8

RequestNotes

Name	Data Type	Minimum Data Size	Maximum Data Size
<i>RequestNotesID</i>	INT	4	4
<i>RequestID</i>	INT	4	4
<i>NotesDescription</i>	VARCHAR(45)	1	46
<i>NotesDateStamp</i>	DATETIME	8	8
<i>isOpen</i>	BOOLEAN	1	1
<i>isComplete</i>	BOOLEAN	1	1
<i>ClosedNotes</i>	VARCHAR(255)	1	256
<i>CompleteNotes</i>	VARCHAR(255)	1	256
	Totals:	21	576

RequestType

Name	Data Type	Minimum Data Size	Maximum Data Size
<i>RequestTypeID</i>	INT	4	4
<i>TopicLabel</i>	VARCHAR(45)	1	46
<i>Application</i>	VARCHAR(45)	1	46
	Totals:	6	96

Grand Totals for Data Tables

		Minimum Data Size	Maximum Data Size
	Grand Totals:	94	1009

As we can see from the above, the system mainly stores numbers and relatively short plaintext ascii strings. Thus the proposed database requires very little storage overhead even in the worst case.

Yearly Storage Cost Estimates

To estimate the yearly data storage requirements for a particular deployment we can make some rough calculations as outlined below (all calculations are in units of bytes):

The storage costs for Users, Departments, and RequestTypes are relatively fixed for a given instance of the database.

Let the number of users of the system be given by N_U , the number of departments be given by N_D , and the number of change request types be given by N_T .

The fixed costs F_C of the above can be estimated by:

$$F_C \approx 200N_U + 100N_D + 200N_T$$

The ChangeRequests themselves, along with their RequestNotes will require varying amounts of storage depending on the frequency of change requests and the volume of notes attached to a given change request.

Let the average number of requests in a year be given by R_Y , and the average number of notes per request be given by N_R .

The yearly storage costs incurred from change requests and notes Y_C can be approximated by:

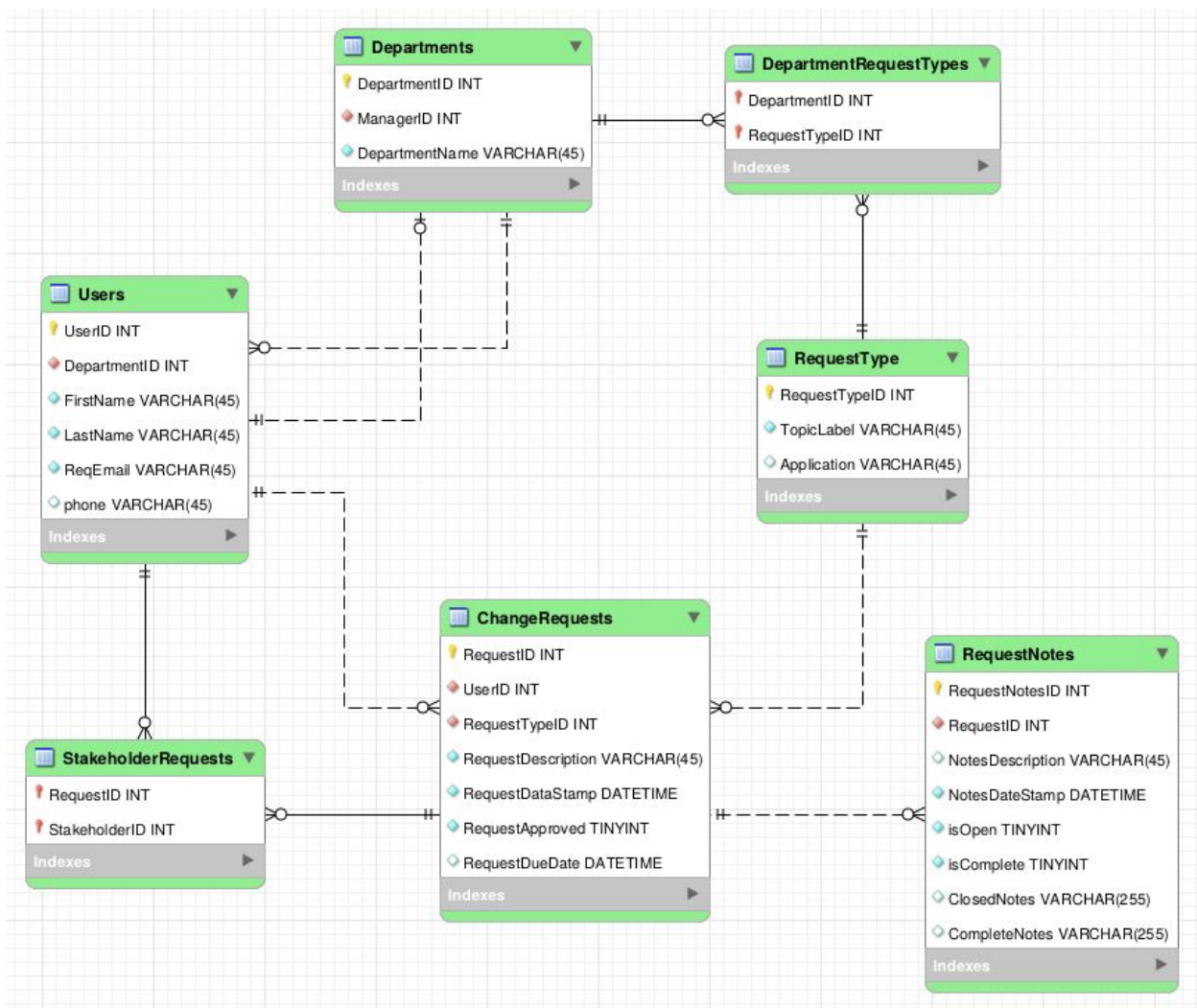
$$Y_C \approx 100R_Y(600N_R)$$

Thus the total storage cost C_T to operate the system for a given number of years N_Y can be given by:

$$C_T \approx F_C + Y_C (N_Y)$$

These “back of the envelope” calculations are conservative, ie. they tend to overestimate the storage costs of the system. This should allow for some flexibility if the system needs to grow slightly. If the system undergoes a large growth then a more careful analysis will be required.

Entity Relationship Diagram



Business Rules

- 1) Departments have zero or more RequestTypes for the Department
- 2) RequestTypes can be used by different Departments
- 3) Departments have exactly one manger (the Manager is also a user)
- 4) If a user is a Manager, they manage exactly one department
- 5) Departments have zero or more Users
- 6) Users belong to exactly one department
- 7) Users can submit zero or more ChangeRequest
- 8) A ChangeRequest is submitted by exactly one User
- 9) ChangeRequests have zero or more stakeholders
- 10) Users can be stakeholders for zero or more ChangeRequests
- 11) ChangeRequests have exactly one RequestType
- 12) ChangeRequest can have zero or more RequestNotes
- 13) RequestNotes belong to exactly one ChangeRequest

