

Название процедуры/функции	Входные параметры	Выходные параметры	Семантическое описание	Объекты, изменяемые процедурой/функцией
show_best			Выбирает 4 верхних отсортированных по типу ID из таблицы Crew_Type строки из таблицы	
check_right_date	@Departure_date, @Logbook_Note, @Logbook_Note_date, @Cabinbook_Note, @Cabinbook_Note_Date	@msg	Дата вылета не должна быть раньше системной даты Если запись в бортовом журнале\кабинном журнале не null, то дата внесения записей в них не должна быть null, а также не должна быть позже системной Иначе дата внесения не может быть не null Переменной @msg присваивается в конце либо фраза об успешности проверки, либо текст ошибки	
check_right_time	@Departure_time, @Logbook_Note, @Logbook_Note_time, @Cabinbook_Note, @Cabinbook_Note_Time, @message	@msg	Функция переходит к проверке только в том случае, если предыдущая функция передала в @msg сообщение об успешном выполнении Время вылета не должно быть раньше системного времени Если запись в бортовом журнале\кабинном журнале не null, то время внесения записей в них не должна быть null, (если дата меньше сегодняшней, часы могут	

			<p>быть какими угодно, если такая же - время должно быть меньше системного)</p> <p>Иначе: время внесения записей не может быть не null</p> <p>Переменной @msg присваивается в конце либо фраза об успешности проверки, либо текст ошибки</p>	
check_right_line	@Tail_number, @Flight_Number, @Logbook_Note, @Logbook_Lastname, @Cabinbook_Note, @Cabinbook_Lastname, @message	@msg	<p>Функция переходит к проверке только в том случае, если предыдущая функция передала в @msg сообщение об успешном выполнении</p> <p>Последовательно проверяется правильность формата бортового номера, рейса ВС и фамилии посредством регулярных выражений</p> <p>Фамилия в кабинном/бортовом журнале может быть прописана, если существует какая-либо запись в соответствующем журнале</p> <p>Переменной @msg присваивается в конце либо фраза об успешности проверки, либо текст ошибки</p>	
Add_Aircraft	@Tail_number varchar(6),		Происходит последовательный вызов	Aircraft Logbook,

	<pre> @Flight_Number   varchar(6), @model   varchar(30), @Departure_date   date, @Departure_time   time, @Logbook_ID   int,  @Logbook_Note_date   date,  @Logbook_Note_time   time, @Logbook_Note   varchar(100),  @Logbook_Lastname   varchar(30), @Cabinbook_ID   int, @Cabinbook_Note   varchar(100),  @Cabinbook_Note_Date   date,  @Cabinbook_Note_Time   time,  @Cabinbook_Lastname   varchar(30) </pre>		<p>функций, проверяющих правильность введенных данных, если все данные корректны, процедура получает сообщение об успешности проверки и осуществляется попытка insert данных в соответствующие таблицы в БД. При возникновении ошибки или если какая то из функций не вернула сообщения об успешности проверки, происходит вывод текста ошибки или системной ошибки с последующем откатом транзакции для предотвращения внесения в таблицы всех (или части) данных</p>	Cabinbook
show_worst		<pre> Post, Lastname, Name_, Patronymic, Flying_Hours, (SELECT AVG(Flying_Hours) FROM Crew), </pre>	<p>Вывод агрегированных и детализированных данных. Вычисляется разница среднего количества часов налета среди всех членов экипажа. Выводятся только</p>	

		Flying_Hours - (SELECT AVG(Flying_Hours))	те, кто имеет отрицательную разницу, т.е. их кол-во часов налета меньше, чем у остальных	
show_deleted	@Day='2020-10-11', @time_1='16:00:00', @time2 = '18:00:00'	Flight_Number, Departure_Date, Departure_Time	Вывод тех самолетов в таблице Aircraft, у которых время и дата вылета находятся в указанном пользователем промежутке. В соответствие этим данным ставится номер рейса из таблицы Logbook	
show_senior	@Flight_number varchar(6)	@Seniors_Lastname	Строится сложный запрос через три таблицы для составления соответствия номера рейса (табл. Logbook) и фамилией сотрудника (табл. Conclusion), который разрешил осуществления вылета самолета (табл. Aircraft) с данным рейсом	